UNIVERSIDADE DO VALE DO ITAJAÍ
CENTRO DE CIÊNCIAS TECNOLÓGICAS DA TERRA E DO MAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

UM SISTEMA ESPECIALISTA PARA AUXÍLIO NA DETECÇÃO DE DEFEITOS EM HARDWARE DE MICROCOMPUTADORES CONSIDERANDO INCERTEZAS

Área de Inteligência Artificial

por

Edson Gil Alves Junior

Raimundo Celeste Ghizoni Teive, Dr.
Orientador

São José (SC), julho de 2008

UNIVERSIDADE DO VALE DO ITAJAÍ
CENTRO DE CIÊNCIAS TECNOLÓGICAS DA TERRA E DO MAR
CURSO DE CIÊNCIA DA COMPUTAÇÃO

UM SISTEMA ESPECIALISTA PARA AUXÍLIO NA DETECÇÃO DE DEFEITOS EM HARDWARE DE MICROCOMPUTADORES CONSIDERANDO INCERTEZAS

Área de Inteligência Artificial

por

Edson Gil Alves Junior

Relatório apresentado à Banca Examinadora do Trabalho de Conclusão do Curso de Ciência da Computação para análise e aprovação.
Orientador: Raimundo Celeste Ghizoni Teive, Dr.

São José (SC), julho de 2008

DEDICATÓRIA

Dedico o meu trabalho a todos que de alguma forma contribuíram para a sua realização, em especial para a minha família que esteve sempre ao meu lado dando forca e apoio em todos os momentos.

AGRADECIMENTOS

Agradeço a minha mãe Estela que por seus esforços pude estudar e seguir com a minha formação profissional e principalmente pessoal, tornando a pessoa que sou hoje.

Agradeço também as minhas irmãs, Giselle que sempre acreditou e me incentivou a buscar aquilo que eu acreditava seja pessoal ou profissionalmente e a Manuella que mesmo ainda não entendendo nada por causa dos seus 2 anos de idade, sei que sempre estará torcendo por mim.

Agradeço a minha namorada Cintia que teve paciência sempre que passava horas estudando para a realização deste projeto, e mesmo assim sempre me incentivou a não desanimar.

Agradeço ao meu padrasto Otto que como meus amigos, estiveram do meu lado durante essa jornada que também não me deixaram desanimar.

Agradeço aos meus professores que de alguma forma conseguiram passar seu conhecimento e experiências que irei lembrar para sempre. Dentre eles destaco meu orientador, Professor Raimundo que acreditou e me incentivou a não desistir do meu trabalho.

Por fim agradeço a todos que de uma forma ou de outra contribuíram para a realização deste trabalho.

SUMÁRIO

LISTA DE ABREVIATURAS v
LISTA DE FIGURAS vi
LISTA DE TABELAS vii
LISTA DE EQUAÇÕES viii
RESUMO ix
ABSTRACT x
1 INTRODUÇÃO 1
1.1 PROBLEMATIZAÇÃO 2
1.1.1 Formulação do Problema 2
1.1.2 Solução Proposta 2
1.2 OBJETIVOS 3
1.2.1 Objetivo Geral 3
1.2.2 Objetivos Específicos 3
1.3 METODOLOGIA 3
1.4 ESTRUTURA DO TRABALHO 4
2 FUNDAMENTAÇÃO TEÓRICA 5
2.1 SISTEMAS ESPECIALISTAS 6
2.1.2 Lógica Nebulosa ou Fuzzy Logic 16
3 DESENVOLVIMENTO 26
3.1 MODELAGEM 26
3.1.1 Metodologia 26
4 RESULTADOS 47
4.1 TESTE 01 47
4.2 TESTE 02 50
4.3 TESTE 03 52
4.4 TESTE 04 54
4.5 TESTE 05 56
4.6 TESTE 06 58
5 CONCLUSÕES 60
5.1 TRABALHOS FUTUROS 61
REFERÊNCIAS BIBLIOGRÁFICAS 62
GLOSSÁRIO 64
A Fluxograma de testes 66
A.1 PROBLEMAS MAIS COMUNS 66

LISTA DE ABREVIATURAS

AC	Aquisição de Conhecimento
AGP	Accelerated Graphics Port
ANEEL	Agência Nacional de Energia Elétrica
BC	Base de Conhecimento
BIOS	Basic Input/Output System
CELESC	Centrais Elétricas de Santa Catarina S.A.
CPU	Central Processing Unit (Unidade Central de Processamento)
DC	Direct Current (Corrente Contínua)
HD	Hard Disk
IA	Inteligência Artificial
MI	Mecanismo de inferências
MT	Memória de Trabalho
PCIe	Peripheral Component Interconnect Express
RAM	Random Access Memory
RGB	Abreviatura do sistema de cores aditivas formadas por Vermelho (Red), Verdes (Green) e Azuis (Blue)
SBC	Sistemas Baseados em Conhecimento
SE	Sistema Especialista
TCC	Trabalho de Conclusão de Curso
UNIVALI	Universidade do Vale do Itajaí
USB	Universal Serial Bus
VDC	Voltage Direct Current (Voltagem de corrente contínua)

LISTA DE FIGURAS

Figura 1. Componentes de um Sistema Especialista....11
Figura 2. Um sistema de Inferência pode ser composto de 5 blocos principais 19
Figura 3. Gráfico fuzzy que representa a temperatura ambiente....22
Figura 4. Fluxograma de Testes 30
Figura 5. Teste da saída 12v....36
Figura 6. Teste da saída 5v....36
Figura 7. Teste da saída 3.3v....37
Figura 8. Teste do cooler....37
Figura 9. Teste do fusível....38
Figura 10. Teste da alimentação 220v AC 39
Figura 11. Saida do teste da Fonte. 39
Figura 12. Exemplo de regra do módulo Fonte.....40
Figura 13. Exemplo de regra do módulo Monitor.....41
Figura 14. Exemplo de regra do módulo CD-ROM.....41
Figura 15. Exemplo de regra do módulo HD. 41
Figura 16. Exemplo de regra do módulo FLOPPY.....41
Figura 17. Exemplo de regra do módulo Placas Off-boards. 42
Figura 18. Exemplo de regra do módulo Placa de Vídeo.....42
Figura 19. Exemplo de regra do módulo Memória. 42
Figura 20. Exemplo de regra do módulo Placa Mãe. 43
Figura 21. Exemplo de regra do módulo Processador. 43
Figura 22. Exemplo de regra do módulo Periférico. 43
Figura 23. Tela principal do FUZZYCLIPS 44
Figura 24. Exemplo de definição de variáveis globais....44
Figura 25. Exemplo de funções criadas pelo usuário....45
Figura 26. Exemplo de definições dos conjuntos difusos 45
Figura 27. Exemplo de regras de startup....45
Figura 28. Exemplo de definições de regra....46
Figura 29. Distribuição de problemas de Hardware....47
Figura 30. Interface Teste01....49
Figura 31. Teste01 de inferência da Fonte. 50
Figura 32. Interface Teste02....51
Figura 33. Teste02 da Fonte. 51
Figura 34. Interface Teste03....53
Figura 35. Teste03 do HD. 54
Figura 36. Interface Teste04....55
Figura 37. Teste04 da Placa de Video. 55
Figura 38. Interface Teste05....57
Figura 39. Teste05 do Processador. 57
Figura 40. Interface Teste06....59
Figura 41. Teste06 da Memória. 59
Figura 42. Fluxograma de testes baseado nos problemas 67

LISTA DE TABELAS

Tabela 1. Definições de Inteligência Artificial. 5
Tabela 2. Exemplo de Regra. 35
Tabela 3. Chamado 01.....48
Tabela 4. Chamado 02.....50
Tabela 5. Chamado 03.....52
Tabela 6. Chamado 04.....54
Tabela 7. Chamado 05.....56
Tabela 8. Chamado 06.....58
Tabela 9. Sequência de testes.....66

LISTA DE EQUAÇÕES

Equação 1 23
Equação 2 23
Equação 3 24
Equação 4 24

RESUMO

ALVES, Edson Gil Jr. **Um Sistema Especialista para Auxílio na Detecção de Defeitos em Hardware de Microcomputadores Considerando Incertezas.** São José, 2008. 74 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Centro de Ciências Tecnológicas da Terra e do Mar, Universidade do Vale do Itajaí, São José, 2008.

Com a popularização do microcomputador e a sua aplicação nas mais diversas áreas do mercado, a necessidade de prestar manutenção de forma rápida e precisa tornou-se indispensável. Contudo, as assistências técnicas muitas vezes contam com profissionais inexperientes que não possuem o conhecimento necessário para atender esta demanda do mercado e ao mesmo tempo permanecerem competitivas. O desenvolvimento de um sistema inteligente para suporte a esta atividade, torna o técnico mais ágil na detecção do problema ali presente. Como muitas vezes a descrição do problema é feita por pessoas que não são da área técnica, como usuários, por exemplo, ou apenas feita de forma vaga ou parcial, esta ferramenta trata estas incertezas da entrada de dados, a fim de apresentar as possíveis causas do problema descrito, de forma ágil e confiável. Para isso foram utilizadas técnicas de Inteligência Artificial (IA) para a modelagem do conhecimento, o qual foi adquirido de técnicos experientes na área. Para tratar as incertezas das informações de entrada, se fez uso de um sistema especialista *fuzzy*. Para a modelagem do sistema foi utilizadas ferramentas computacionais como a Shell FUZZYCLIPS, que é um Shell para desenvolvimento de Sistemas Especialistas baseados em lógica difusa, bastante consagrado no meio acadêmico.

Palavras-chave: Sistema Especialista. Inteligência Artificial. Lógica difusa.

ABSTRACT

With the popularization of the microcomputer and its application in areas of the market, the need to provide quickly and accurately maintenance has become indispensable. However, the technical assistance often have inexperienced professionals who do not have the necessary knowledge to suply the demand of the market, and, at the same time to continue competitive. The development of an intelligent system to support this activity, helps the technician to be more agile in the detection of the micro computer's problem. But, often the problem's description is made by people who are not used to the technical area, such as users, or the description is done in a vague way. This tool deal with these uncertainties of the input data in order to present the possible causes of the problem described, in an agile and reliable way. To do so, it was used techniques of Artificial Intelligence (AI) for the modeling of knowledge, each was acquired by experienced technicians of the area. To deal with the uncertainties of the input information, the fuzzy logic was used, the system as a whole, was developed as a fuzzy expert system. For the modeling of the system was used computational tools such as Shell FUZZYCLIPS, a Shell for the development of Expert Systems based on fuzzy logic, common use in academic research

***Keywords**: Expert Systems. Artificial Intelligence. Fuzzy Logic.*

1 INTRODUÇÃO

Atualmente, as mais diversas áreas exploram o uso da informática em suas atividades, seja por praticidade, agilidade ou até mesmo pela flexibilidade que o microcomputador oferece para os seus usuários. Com essa popularização, a necessidade de prestar uma manutenção rápida e eficiente tornou-se indispensável para qualquer empresa. Estas, por sua vez, contam muitas vezes com técnicos sem muita experiência profissional e que necessitam de mais tempo para passar um diagnóstico confiável do problema para o cliente.

O desenvolvimento de um Sistema Especialista (SE) com lógica difusa (*fuzzy logic*) que auxilie técnicos em manutenção de microcomputadores no diagnóstico de problemas de hardware, tornaria esta tarefa mais rápida e confiável. Para um mercado concorrido, como da assistência técnica, a agilidade na prestação do serviço é fundamental.

Segundo Savaris (2002), os SEs procuram capturar e simular o conhecimento de especialistas humanos. Eles são desenvolvidos para tratar problemas complexos do mundo real que necessitem a interpretação e a análise de especialistas humanos e ao mesmo tempo, cheguem a conclusões e decisões similares ao fornecido pelo especialista humano.

Um SE com Lógica Difusa é aplicado para esse tipo de problema devido ao seu funcionamento baseado em encadeamento de regras, embora não determinístico (100% de certeza). Isto é devido à entrada de informações que muitas vezes é feita de forma vaga, tal como um técnico que baseado nos sintomas, utilizaria seus conhecimentos na resolução do problema.

Sabe-se que a entrada de dados é incerta, pois na maioria das vezes quem informará os sintomas do problema ao técnico, é uma pessoa que não possui o conhecimento técnico. Ainda, os sintomas podem ser passados de forma incompleta, dificultando ainda mais o seu diagnóstico. A Lógica Difusa conhece estas incertezas, ou seja, os “valores verdades” são expressos lingüisticamente, como:

- Quente, frio, muito frio, muito quente;
- Longe, perto;
- Rápido, vagaroso;
- Grande, médio, pequeno, etc.

Também possui vários modificadores de predicado como:

- Muito, mais ou menos, pouco, bastante, etc.

Possui também um amplo conjunto de quantificadores, como:

- Poucos, vários, em torno de, usualmente.

Faz uso das probabilidades lingüísticas, como, provável, improvável, que são interpretados como números *fuzzy* e manipulados pela sua aritmética. Manuseia todos os valores entre 0 e 1, tomando estes, como um limite apenas, tornando assim a interação usuário máquina mais amigável.

1.1 PROBLEMATIZAÇÃO

1.1.1 Formulação do Problema

O diagnóstico de defeito de hardware em microcomputadores, envolve a consideração de vários aspectos técnicos e alternativas possíveis de solução. Mesmo um técnico experiente pode esquecer detalhes relevantes da solução. Algumas vezes, não se tem 100% de certeza sobre os sintomas e eventuais estados dos componentes que compõe o hardware do mesmo, justificando a necessidade de uma técnica que possa modelar esta incerteza. Por isso é importante o desenvolvimento de sistemas computacionais, que possam tratar a incerteza referente ao problema, possibilitando o diagnóstico do defeito com melhor confiabilidade.

1.1.2 Solução Proposta

O desenvolvimento de um Sistema Especialista *Fuzzy* para o auxílio na detecção de problemas de hardware, sugerindo assim uma solução, seja a substituição do componente ou a realização de algum procedimento corretivo.

A utilização de um SE tradicional não foi adotada, uma vez que a entrada de dados é incerta, tornando inviável a criação de uma base de regras e de fatos puramente determinísticas. Uma solução baseada em SE tradicional, para esta aplicação, certamente teria um custo computacional (processamento) superior a um SE *fuzzy* devido à quantidade superior de regras.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho tem com objetivo o desenvolvimento de um sistema especialista *fuzzy* para auxiliar no diagnóstico de defeitos de hardware em microcomputadores, para análise do técnico em manutenção.

1.2.2 Objetivos Específicos

- Estudar a extensão da Shell de sistema especialista CLIPS, o FUZZYCLIPS;
- Estudar e analisar bibliografia sobre problemas de hardware em microcomputadores;
- Realizar a aquisição do conhecimento necessária sobre hardware;
- Implementar a base de conhecimento do sistema;
- Projetar e implementar um sistema especialista *fuzzy*; e
- Testar e validar a implementação do sistema.

1.3 METODOLOGIA

Para o presente trabalho foi adotada a metodologia que consistiu inicialmente na revisão bibliográfica sobre duas grandes áreas de pesquisa: Inteligência Artificial e manutenção de computadores. Na primeira, foram aprofundadas técnicas como: Sistemas Especialistas, Sistemas Especialistas *Fuzzy* e Problemas (diagnósticos). Na segunda área de estudo foi estudada a parte de manutenção de microcomputadores e defeitos mais comuns em hardware de microcomputadores. Foi realizado um estudo focado sobre lógica *fuzzy* através de livros, apostilas e implementação no software MATLAB 6.5. Outra etapa consistiu na análise da *Shell* FUZZYCLIPS, envolvendo a utilização de apostilas, livros e implementação de exemplos. A aquisição do conhecimento foi feita na quarta etapa, através de entrevistas com especialistas da área, onde foram levantados os problemas mais comuns encontrados em microcomputadores. A representação do conhecimento foi realizada na quinta etapa. A sexta e sétima etapa realizada consistiu a implementação do sistema especialista *fuzzy* e sua validação através de testes respectivamente. Nos testes foram simulados problemas reais, descritos por especialistas para uma posterior verificação das soluções sugeridas, uma vez o resultado dos testes positivos, o sistema estará validado.

1.4 ESTRUTURA DO TRABALHO

Este documento está estruturado em quatro capítulos. O Capítulo 1, Introdução, apresenta um panorama geral do trabalho, assim como as ferramentas utilizadas em sua confecção. No Capítulo 2, Fundamentação Teórica, é apresentada uma revisão bibliográfica sobre: Inteligência Artificial, Sistemas Especialistas, Lógica *Fuzzy*, assim como uma análise dos problemas mais comuns encontrados em hardware e um estudo da ferramenta FUZZYCLIPS. O Capítulo 3 apresenta o projeto, detalhando do sistema desenvolvido, incluindo sua especificação e a suas limitações. O Capítulo 4 apresenta alguns exemplos de simulações utilizadas para a validação do sistema. No Capítulo 5, apresentam-se as considerações finais, onde são abordados os resultados, e sugestão de trabalhos futuros. Por fim têm-se as referências bibliográficas utilizadas na realização deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados alguns conceitos preliminares necessários para o desenvolvimento deste trabalho. Será apresentado ainda, um aprofundamento no campo específico da Inteligência Artificial: Sistemas Especialistas com Lógica Difusa.

Atualmente, a IA abrange uma enorme variedade de subcampos, desde áreas de uso geral, como aprendizado e percepção, até tarefas especificas como jogos de xadrez, demonstração de teoremas matemáticos, criação de poesia e diagnóstico de doenças. A IA sistematiza e automatiza tarefas intelectuais e, portanto, é potencialmente relevante para qualquer esfera da atividade intelectual humana. Nesse sentido, ela é verdadeiramente um campo universal (RUSSEL & NORVIG, 2004).

As definições de Inteligência Artificial de acordo com seis livros didáticos são mostradas na Tabela 1.

Tabela 1. Definições de Inteligência Artificial.

Sistemas que pensam como seres humanos	**Sistemas que pensam racionalmente**
"O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal." (HAUGELAND, 1985, *apud* RUSSEL & NORVIG, 2004)	"O estudo das faculdades mentais pelo uso de modelos computacionais." (CHARNIAK & MCDERMOTT, 1985, *apud* RUSSEL & NORVIG, 2004)
"Automatização de atividades que associamos aos pensamentos humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado..." (BELLMAN, 1978, *apud* RUSSEL & NORVIG, 2004)	"O estudo das computações que tornam possível, perceber, raciocinar e agir." (WINSTON, 1992, *apud* RUSSEL & NORVIG, 2004)
Sistemas que atuam como seres humanos	**Sistemas que atuam racionalmente**

“A arte de criar máquinas que executa, funções que exigem inteligência quando executadas por pessoas”.(KURZWEIL, 1990, *apud* RUSSEL & NORVIG, 2004)	“A Inteligência Computacional é o estudo do projeto de agentes inteligentes”.(POOLE et al., 1998, *apud* RUSSEL & NORVIG, 2004)

Essas definições variam ao longo de duas dimensões principais. Em linhas gerais, as que estão na parte superior da tabela se relacionam a processos de pensamentos e raciocínio, enquanto as definições da parte inferior se referem ao comportamento. As definições do lado esquerdo medem o sucesso em termos de fidelidade ao desempenho humano, enquanto as definições do lado direito medem o sucesso comparando-o a um conceito ideal de inteligência, que chamaremos de racionalidade. Um sistema é racional se “faz tudo certo”, com os dados que tem.

Historicamente, todas as quatro estratégias para estudo da IA têm sido seguidas. Como se poderia esperar existe uma tensão entre abordagens centradas em torno de seres humanos e abordagens centradas em torno da racionalidade. Uma abordagem centrada nos seres humanos deve ser uma ciência empírica, envolvendo hipóteses e confirmação experimental. Uma abordagem racionalista envolve uma combinação de matemática e engenharia. Cada grupo tem ao mesmo tempo desacreditado e ajudado o outro (RUSSEL & NORVIG, 2004).

2.1 Sistemas Especialistas

Conforme Passos (1989), este subgrupo da IA é assim denominado porque possui conhecimento de um especialista em uma determinada área. Tem como objetivo resolver os problemas relativos a um domínio específico, podendo ser aplicado a praticamente todas as áreas que precisem de um especialista.

Passos (1989), afirma ainda, que um sistema especialista é um programa inteligente de computador que utiliza métodos e inferências para a resolução de problemas técnicos e altamente especializados. Os sistemas especialistas interagem com seu usuário numa linguagem natural de perguntas e respostas, sugerindo e auxiliando na solução de problemas complexos.

Programas de computador que tentam resolver problemas que os seres humanos resolveriam emulando o raciocínio de um especialista, aplicando conhecimentos específicos e inferências são ditos “Sistemas Especialistas” (SAVARIS, 2002).

Em um sistema convencional, seu funcionamento é baseado em um algoritmo, que emite um resultado final ao processar um volume de dados de maneira repetitiva, um sistema especialista é baseado em uma busca heurística que trabalha com problemas para os quais não existe uma solução convencional organizada de forma algorítmica disponível ou é muito demorada (SAVARIS, 2002).

De acordo com Ribeiro (1987), um sistema especialista é aquele que é projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano. É capaz de emitir uma decisão, com apoio em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano. Um sistema especialista deve, além de inferir conclusões, ter capacidade de aprender novos conhecimentos e, desse modo, melhorar o seu desempenho de raciocínio, e a qualidade de suas decisões.

Para tomar uma decisão sobre um determinado assunto, um especialista o faz a partir de fatos que encontra e de hipóteses que fórmula, buscando em sua memória um conhecimento prévio armazenado durante anos, no período de sua formação e no decorrer de sua vida profissional, sobre esses fatos e hipóteses. E o faz de acordo com a sua experiência, isto é, com o seu conhecimento acumulado sobre o assunto e, com esses fatos e hipóteses, emite a decisão (SAVARIS, 2002).

Durante o processo de raciocínio, o mecanismo de inferência vai verificando qual a importância dos fatos que encontra, comparando-os com as informações já contidas no seu conhecimento acumulado sobre esses fatos e hipóteses. Neste processo, vai formulando novas hipóteses e verificando novos fatos, que influenciaram no processo de raciocínio. Este raciocínio é sempre baseado no conhecimento prévio acumulado. Um especialista com esse processo de raciocínio pode não chegar a uma decisão se os fatos de que dispõe para aplicar o seu conhecimento prévio não forem suficientes. Pode, por este motivo, inclusive chegar a uma conclusão errada, mas este erro é justificado em função dos fatos que encontrou e do seu conhecimento acumulado previamente (SAVARIS, 2002).

Segundo Harmon e King (1988), SE é um programa inteligente de computador que usa conhecimentos e inferências, para resolver problemas que são bastante difíceis, de forma a requererem para sua solução, muita perícia humana. O conhecimento necessário para atuar a esse nível, mais os procedimentos inferências empregados, pode-se considerar um modelo da perícia aos melhores profissionais do ramo. O conhecimento de um sistema especialista consiste em fatos e heurísticas. Os fatos constituem um corpo de informação que é largamente compartilhado, publicamente disponível e geralmente aceito pelos especialistas em um campo. As heurísticas são,

em sua maioria privadas, regras poucos discutidas, de bom discernimento (regras de raciocínio plausível, regras de boa conjectura), que caracterizam a tomada de decisão ao nível de especialista na área. O nível de desempenho de um sistema especialista é função principalmente do tamanho e da qualidade do banco de conhecimento que possui.

Para Farreny (1985, *apud* SAVARIS, 2002), um sistema só é considerado especialista se ele possui as seguintes características:

- Uma linguagem de expressão dos conhecimentos fornecidos pelos especialistas;
- Uma base de conhecimentos, para armazenar o conhecimento específico de determinada aplicação, que pode ser diretamente fornecido por um especialista, ou acumulado pelo sistema ao fim dos experimentos; e
- Um motor de inferência, programa que explora o conhecimento da base precedente, considerando-a como fonte de informações.

Os sistemas especialistas são softwares que os peritos em campos específicos enriquecem com seu conhecimento, capacitando um computador a auxiliá-lo num processo de tomada de decisão. A intenção em usar conhecimento simbólico para simular o comportamento dos especialistas humanos, é justamente, desenvolver programas que possibilitem a utilização dos conhecimentos dos especialistas através de uma máquina que permita o armazenamento e o seqüenciamento de informações e a auto-aprendizagem.

Assim Chorafas (1988), coloca que: "o computador interage com as perguntas do usuário e chega a uma conclusão baseada nas respostas. O usuário pode perguntar ao computador por que ele quer certa informação e o SE explicará sua necessidade dos dados e como eles serão utilizados. O mais importante é que o SE dirá como chegou a suas conclusões; não dará somente conselhos, mas também justificará a opinião que oferece".

Segundo Carvalho, Monteiro e Genaro (1987), para entender quais são as características comuns aos Sistemas Especialistas, basta examinar o que estes fazem:

- Resolvem problemas muito complexos tão bem quanto ou melhor que especialistas humanos, pois as pessoas estão suscetíveis a influências externas, emocionais, culturais, etc.;

- Raciocinam heuristicamente, usando o que os peritos consideram efetivamente regras práticas;
- Interagem com usuários humanos utilizando inclusive linguagem natural;
- Manipulam e raciocinam sobre descrições simbólicas;
- Funcionam com dados errados e regras incertas de julgamento;
- Contemplam hipóteses múltiplas simultaneamente;
- Explicam porque estão fazendo determinadas pergunta; e
- Justificam suas conclusões.

Logo um sistema especialista procura simular o conhecimento de um especialista humano de determinada área, e baseado em regras, resolver os problemas como os especialistas resolveriam.

2.1.1.1 Classificação dos Sistemas Especialistas

Segundo Rabuske (1995), pode-se classificar os Sistemas Especialistas quanto às características do seu funcionamento. De um modo geral, tais categorias são:

- **Interpretação** - São sistemas que inferem descrições de situações a partir da observação de fatos fazendo uma análise de dados e procurando determinar as relações e seus significados. Devem considerar as possíveis interpretações, descartando as que se mostrarem inconsistentes.
- **Diagnósticos** - São sistemas que detectam falhas oriundas da interpretação de dados. A análise dessas falhas pode conduzir à uma conclusão diferente da simples interpretação de dados. Detectam os problemas mascarados por falhas dos equipamentos e falhas do próprio diagnóstico, que este não detectou por ter falhado. Estes sistemas já têm embutido o sistema de interpretação de dados.
- **Monitoramento** - Interpreta as observações de sinais sobre o comportamento monitorado. Tem de verificar continuamente um determinado comportamento em limites pré-estabelecidos, sinalizando quando forem requeridas intervenções para o sucesso da execução. Um sinal poderá ser interpretado de maneiras diferentes, de acordo com a situação global percebida naquele momento, e a interpretação varia de acordo com os fatos que o sistema percebe a cada momento.
- **Predição** - A partir de uma modelagem de dados do passado e do presente, este sistema permite uma determinada previsão do futuro. Como ele baseia sua solução na análise do comportamento dos dados recebidos no passado, de ter mecanismos

para verificar os vários futuros possíveis, a partir da análise do comportamento desses dados, fazendo uso de raciocínios hipotéticos e verificando a tendência de acordo com a variação dos dados de entrada.

- **Planejamento** - Neste caso, o sistema prepara um programa de iniciativas a serem tomadas para se atingir um determinado objetivo. São estabelecidas etapas e subetapas e, em caso de etapas conflitantes, são definidas as prioridades. Possui características parecidas com o sistema para a predição e normalmente opera em grandes problemas de solução complexa. O princípio de funcionamento, em alguns casos, é por tentativas de soluções, cabendo a análise mais profunda ao especialista que trabalha com esse sistema. Enfoca os aspectos mais importantes e particiona de maneira coerente um problema em subproblemas menos complexos, estabelecendo sempre o relacionamento entre as metas destes subproblemas e a meta principal.
- **Projeto** - Este sistema tem características parecidas com as do planejamento, e devem-se confeccionar especificações tais que sejam atendidos os objetivos dos requisitos particulares. É um sistema capaz de justificar a alternativa tomada para o projeto final, e de fazer uso dessa justificativa para alternativas futuras.
- **Depuração** - Trata-se de sistemas que possuem mecanismos para fornecerem soluções para o mau funcionamento provocado por distorções de dados. Provê, de maneira automática, verificações nas diversas partes, incluindo mecanismos para ir validando cada etapa necessária em um processo qualquer.
- **Reparo** - Um sistema especialista para reparos segue um plano para administrar alguma solução encontrada em uma etapa do diagnóstico. São poucos os sistemas desenvolvidos, porque o ato de executar um conserto em alguma coisa do mundo real é uma tarefa complexa.
- **Instrução** - O sistema de instrução tem um mecanismo para verificar e corrigir o comportamento do aprendizado dos estudantes. Normalmente, incorporam como subsistemas um sistema de diagnóstico e de reparo, e tomam por base uma descrição hipotética do conhecimento do aluno. Seu funcionamento consiste em ir interagindo com o treinamento, em alguns casos apresentando uma pequena explicação e, a partir daí, ir sugerindo situações para serem analisadas pelo treinando. Dependendo do comportamento deste, se vai aumentando a complexidade das situações e encaminhando o assunto, de maneira didática, até o nível intelectual do treinamento.

- **Controle** - É um sistema que governa o comportamento geral de outros sistemas (não apenas de computação). É o mais completo, de um modo geral, pois deve interpretar os fatos de uma situação atual, verificando os dados passados e fazendo uma predição do futuro. Apresenta os diagnósticos de possíveis problemas, formulando um plano ótimo para sua correção. Este plano de correção é executado e monitorado para que o objetivo seja alcançado.

Demonstrando assim a grande variedade de aplicações onde pode ser empregado o uso de sistemas especialistas.

2.1.1.2 Componentes Principais de um Sistema Especialista

Tipicamente, um SE possui os seguintes componentes principais: base de conhecimento (base de regras), mecanismo de inferências e memória de trabalho (base de fatos), conforme é apresentado na Figura 1.

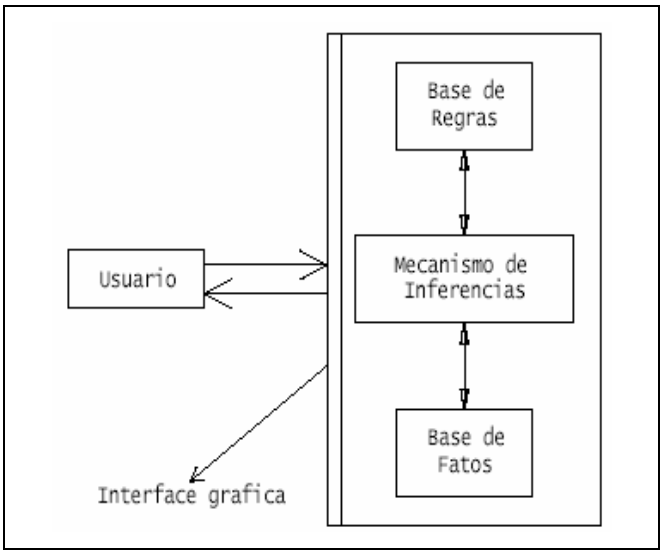


Figura 1. Componentes de um Sistema Especialista
Fonte: Teive (1997).

A base de conhecimento (BC) é o elemento que armazena o conhecimento abstrato. Normalmente, a BC é conhecida como base de regras onde o conhecimento abstrato é armazenado em um conjunto de regras de produção do tipo:

Se (situação) $\Rightarrow$ Então (conclusão ou ação).

A memória de trabalho (MT) é o elemento que armazena o conhecimento concreto, ou seja, o conhecimento que pode ser considerado fato antes do processo de inferenciação. Esta memória é de caráter transitório, pois, novos fatos estão sendo acrescentados continuamente ou fatos existentes são excluídos. A MT é também conhecida como base de fatos (TEIVE, 1997).

O mecanismo de inferências (MI) é o processador ou interpretador de conhecimento, sendo considerado o coração do SE (CASTILHO & ALVAREZ, 1991). A sua principal função é combinar o conhecimento abstrato contido na base de regras, com o conhecimento concreto armazenado na base de fatos, inferindo conclusões e gerando novos fatos.

Segundo Teive (1997), o desenvolvimento de um SE se inicia com a consulta de algumas fontes de conhecimento, tais como: livros, manuais, relatórios técnicos e principalmente da experiência e conhecimento dos especialistas.

2.1.1.3 Representação do conhecimento

A frase “Conhecimento é poder” ilustra a relevância do conhecimento para um SE, segundo Durkin (1994), conhecimento é um termo abstrato que tenta capturar como um indivíduo entende um determinado assunto.

Já pela visão de Fishler e Firschein (*apud* RABUSKE, 1995), conhecimento pode ser definido como a informação armazenada, ou o modelo usado pela pessoa ou máquina para interpretar, predizer e responder apropriadamente ao mundo exterior.

O conhecimento é o que define a importância do sistema, sendo assim de suma importância para um SE. Como existem vários tipos de conhecimento, existem várias formas de representá-los.

Segundo Durkin (1994), o conhecimento pode ser classificado em: Processual, Declarativo, Meta-Conhecimento, Heurístico e Estrutural. Giarrantano e Rilley (1993), também reconhecem a forma Tácita de conhecimento.

- Conhecimento Processual: descreve como o problema é resolvido. Regras, estratégias e procedimentos são os tipos de conhecimento processual usado em SEs.

- Conhecimento Declarativo: descreve o que é conhecido sobre um problema. Demonstra algo que pode ser verdadeiro ou falso. Também inclui uma lista de declarações que descreve com maior suficiência algum objeto ou conceito.

- Meta-Conhecimento: Descreve o conhecimento sobre outro conhecimento. É usado para selecionar outros conhecimentos a fim de aumentar a eficiência da resolução do problema, direcionando o seu conhecimento para áreas mais promissoras.

- Conhecimento Heurístico: Descreve regras que guiam o processo de raciocínio. É empírico e representa o conhecimento de um especialista para resolver um problema, baseado em problemas anteriores.

- Conhecimento Estrutural: descreve o conhecimento global de da mente de um especialista.

- Conhecimento Tácito: refere-se ao conhecimento inconsciente, em Redes Neurais utiliza-se conhecimento tácito em sua representação, porque muitas vezes a mesma não pode explicar seu conhecimento diretamente.

Demonstrando assim a importância do engenheiro de conhecimento para a formação da base de fatos.

2.1.1.4 Técnicas Para Aquisição do Conhecimento

Segundo Mastela (2004), a Aquisição do Conhecimento pode ser definida como o processo de compreender e organizar o conhecimento de várias fontes. Esse conhecimento deverá, então, ser codificado e armazenado em uma BC para que possa ser resgatado pelo SE quando necessário.

O engenheiro de conhecimento é a pessoa responsável por todas as funções de construção de um SE. Ele deve entender o domínio de aplicação do sistema para que possa interagir com as fontes de informações (conhecimento que será incorporado ao SE) e, ainda, deve possuir um conhecimento razoável de computação, linguagens e ferramentas de inteligência artificial, a fim de escolher o melhor para implementação do sistema (COSTA & SILVA, 2005).

Essa dificuldade se dá em virtude da inexistência de uma metodologia eficiente, confiável e padrão para extração e organização do conhecimento das várias fontes. Essa dificuldade existe de fato, no entanto a AC é essencial, pois apenas ela permite a construção de uma BC, que é inerente à existência de um SE (COSTA & SILVA, 2005).

Segundo Costa e Silva (2005), as fontes de conhecimento usadas no processo de AC podem ser documentadas ou não. As fontes documentadas são livros, filmes, gráficos, diagramas, manuais,

entre outras. Já as fontes não documentadas residem na mente das pessoas que atuam no domínio de aplicação do sistema.

Com a utilização de documentos como fonte de dados o engenheiro de conhecimento pode a vir a ter problemas para abstrair as informações necessárias quanto ao domínio do problema. Por exemplo, não é possível para um leigo adquirir um livro de cardiologia e conseguir extrair conhecimentos importantes. Este problema pode vir a ser contornado com a utilização de agentes do domínio, que são pessoas que trabalham diretamente com o assunto em questão.

Segundo Teive (1997), o processo de AC, via engenheiro de conhecimento, envolve uma inter-relação e interação entre o engenheiro e o especialista, sendo que, em muitos métodos o especialista faz o papel de um professor, enquanto que o engenheiro representa o papel do pupilo. Neste sentido, alguns fatores cognitivos tais como: diferenças individuais, compreensão de conceitos, pré-disposição, expectativas e pré-concepções podem afetar a aprendizagem do engenheiro de conhecimento.

Outra dificuldade é que em alguns casos o especialista sabe resolver um problema, mas não sabe explicar como ou por que resolveu daquela maneira. Isso ocorre porque o conhecimento utilizado para resolver o problema é tácito, ou seja, é um conhecimento implícito em sua mente que para ser relatado deve ser explicitado. Há ainda o fato de que algumas pessoas sentem-se avaliadas no processo de entrevistas para AC e acabam omitindo algumas informações importantes para a construção da BC (COSTA & SILVA, 2005).

Costa e Silva (2005), afirmam que muitos esforços têm sido feitos para sistematizar ou até mesmo automatizar o processo de AC. Esses esforços resultaram em várias técnicas, as quais podem ser classificadas em manuais, semi-automáticas e automáticas. As técnicas manuais são as mais utilizadas, as semi-automáticas geralmente são utilizadas em conjunto com as manuais, já as automáticas dizem respeito ao processo pelo qual o conhecimento é adquirido automaticamente, ou seja, sem a interferência humana, ou com pouca interferência. Como exemplo de AC automático temos a mineração de dados e o aprendizado de máquina (redes neurais, árvores de decisões, entre outros).

Técnicas Manuais

Segundo Rezende (2002), a maioria das técnicas manuais fundamenta-se na psicologia e análise de sistemas, sendo o engenheiro de conhecimento o responsável por adquirir o conhecimento do especialista e de outras fontes para codificá-lo na BC:

- **Baseadas em Descrições ou Imersão na Literatura:** Nesta técnica o engenheiro de conhecimento (EC) faz um pequeno estudo sobre o assunto do problema em algumas literaturas com o objetivo de adquirir algum conhecimento inicial sobre o domínio. Isso é importante para que nas futuras entrevistas com o Especialista (ES) a conversa possa fluir em um nível um pouco mais adiantado, pois o simples uso de alguns jargões pelo ES poderá vir a impossibilitar o EC de entender determinadas informações;

- **Baseadas em Entrevistas:** Nesta técnica são realizadas entrevistas com o ES. As informações podem ser coletadas com o auxílio de gravadores, filmadoras, questionários, ou qualquer outro meio possível. Essas informações são posteriormente analisadas para se extrair o conhecimento desejado; e

- **Baseadas em Acompanhamento:** Este método consiste em acompanhar o processo de raciocínio do ES em casos reais. Assim, são pesquisados casos em prontuários ou antigos projetos resolvidos anteriormente pelo especialista. Isso estimula a descrição do ES sobre o assunto em questão. Esta técnica, por utilizar casos reais, evita que o ES seja direcionado a responder questões irrelevantes, no entanto, nem sempre se consegue uma amostragem de casos realmente representativa.

Técnicas Semi-automáticas

As técnicas de AC manuais podem ser bastantes problemáticas devido ao grande número de pessoas envolvidas – especialistas, engenheiros e programadores (REZENDE, 2002). Esse processo de adquirir o conhecimento do especialista e transmitir aos programadores acaba por gerar ruídos de comunicação entre as partes. Uma alternativa para minimizar esse tipo de problema é a aquisição de conhecimento semi-automático, que consiste na utilização de ferramentas computacionais que auxiliam o engenheiro de conhecimento a codificar a BC. Além de reduzir o contingente de pessoas envolvidas, e por conseqüência os problemas de comunicação, a aquisição semi-automática também

acelera o processo de construção da Base de Conhecimento, permitindo que o engenheiro e o especialista tenham respostas mais rápidas, pois à medida que a Base vai sendo construída ela pode ser testada e avaliada assim possíveis erros serão detectados prematuramente.

Os SEs surgem então como uma solução para suprir as necessidades em um mundo que necessita de informações cada vez mais precisas e rápidas. O sucesso de um sistema dessa natureza depende de uma coleta fiel das informações essenciais ao domínio de interesse (COSTA & SILVA, 2005).

2.1.2 Lógica Nebulosa ou Fuzzy Logic

2.1.2.1 Introdução

O termo *fuzzy* em língua inglesa pode ter diversos significados, de acordo com o contexto empregado, mas o conceito básico deste termo é traduzido como: vago, indistinto, incerto, nebuloso. As tentativas de tradução para o português ainda não são uma unanimidade entre os pesquisadores da área: nebuloso e difuso, são os exemplos mais populares, também podem ser encontrados como: lógica “borrosa”, em espanhol (REZENDE, 2002).

Novas tecnologias são inventadas devido a necessidades específicas. O advento da lógica *fuzzy* foi causado pela necessidade de um método capaz de expressar de uma maneira sistemática quantidades imprecisas, vagas, mal-definidas (SHAW & SIMÕES, 1999).

Segundo Brandão (2005), em Inteligência artificial, a lógica *fuzzy* é utilizada para a resolução de uma variedade de problemas, principalmente os relacionados com controle de processos industriais complexos e sistemas de tomada de decisão em geral. Esses sistemas são geralmente tolerantes a imprecisões e ruídos nos dados de entrada e consistem na aplicação da lógica *fuzzy* com a intenção de imitar o pensamento humano na programação de computadores. Com a lógica convencional, os computadores podem manipular valores estritamente binários, como verdadeiro ou falso, sim ou não, 1 (um) ou 0 (zero), ligado ou desligado, etc. Na lógica *fuzzy*, se utilizam de modelos matemáticos para mapear variáveis subjetivas, como frio, agradável e quente, para valores concretos que podem ser manipulados matematicamente.

Conforme Shaw e Simões (1999), operadores humanos, treinados, podem trabalhar com plantas industriais não completamente compreendidas, e processos mal-definidos e sistemas com dinâmicas desconhecidas. Esses operadores sabem qual “ação” tomar, quando observam certas

“condições”, tais como uma combinação de leitura de instrumentos, padrões indicados por sinais luminosos ou sonoros, ou outros eventos. Assim a vantagem de controladores inteligentes é permitir que regras heurísticas possam capturar tais estratégias de controle de operadores humanos. Isso resolve o problema de se automatizar funções de controle geralmente delegadas para controle manual.

A lógica *fuzzy* é também útil na tarefa de tomada de decisões, onde as variáveis de decisões não são definidas em termos exatos. Um fator interessante a se notar é que a lógica *fuzzy* não tem sido aceita facilmente, porque a conceituação utilizada nos princípios da lógica *fuzzy* parece ser contraria as tradições e cultura ocidentais, as quais são baseadas em uma definição precisa, bivalente, entre o ser ou o não ser, o claro e o escuro. Embora a cultura ocidental, fundamentada na lógica binária, resolva de forma extraordinária muitos problemas, é necessário que se preencha os espaços não adequadamente endereçados pelos métodos tradicionais. Nesse sentido, a lógica *fuzzy* permitiria ver “graus de verdade” entre ser e não ser “graus de cinza” entre o claro e o escuro (SHAW & SIMÕES, 1999).

Segundo Bittencourt (1998), lógica *fuzzy* é a forma mais tradicional para o tratamento da informação imprecisa e vaga. Este modelo, introduzido por Zadeh (1978), tem por objetivo permitir graduação na pertinência de um elemento de uma determinada classe, ou seja, de possibilidade de um elemento, pertencer com maior ou menor intensidade a esta classe.

Lógica *fuzzy*, redes neurais, sistemas especialistas e os algoritmos genéticos, pertencem a esse novo paradigma chamado coletivamente por “sistemas inteligentes”. Como esperado de qualquer tecnologia nova tem seus proponentes e seus oponentes (SHAW & SIMÕES, 1999).

2.1.2.2 Princípios Básicos

Bivalência

Segundo Shaw e Simões (1999), o atributo de bivalência significa a utilização de dois valores: algo é verdadeiro ou não-verdadeiro, branco ou preto, ou é um ou zero. A lógica clássica de Aristóteles fez do atributo da bivalência um marco histórico em nossa cultura ocidental. Espera-se sempre que uma determinada afirmação seja verdadeira ou falsa. Não há nada entre ambas, o meio é excluído. A ciência da computação, ou informática, é baseada na bivalência. A álgebra booleana é considerada como uma ferramenta, que permite definir “as leis da verdade” em linguagem matemática. Estes conceitos têm se tornados inter-relacionados e considerados como parte

indispensável do método cientifico: se algo não é absolutamente correto, então não é verdade. Duas vezes dois é quatro, absolutamente. No passado as pessoas costumavam rir das primeiras máquinas de calcular que resultavam em 3.9999999!

Multivalência

Shaw e Simões (1999), afirmam que há um considerável descompasso entre o mundo real e a visão bivalente do mesmo, a começar pelo fato de que o mundo real tem um número infinito de sombreamento e graus de cinza entre as cores preta e branca. Parece que no mundo real é tudo uma questão de ponto de vista ou de graduação, ou seja, tudo depende. O mundo real não é bivalente, é na realidade multivalente com um infinito espectro de opções em vez de duas. Em termos técnicos, o mundo real é analógico, não digital, com muitos tons de cinza entre o branco e o preto. Verdade absoluta e previsão existem apenas como "casos extremos", a insistências nos extremos como validas deve-se apenas a uma polarização cultural, nada mais. Assim, o objetivo da lógica *fuzzy* é o de capturar esses tons de cinza e graus de verdade. A lógica *fuzzy* trabalha, com tal incerteza e verdade parcial os fenômenos naturais, de uma forma sistemática e rigorosa.

Por outro lado, os computadores podem apenas "raciocinar" de forma bivalente: 0 e 1. Os computadores não conseguem entender os termos *fuzzy* da comunicação humana. A lógica *fuzzy* pode preencher esse vazio e traduzir os graus de verdade das afirmações de uma maneira que os computadores possam processar tal informação. A lógica *fuzzy* pode fazer com que os computadores "raciocinem" conforme humanos (SHAW & SIMÕES, 1999).

2.1.2.3 Sistemas Fuzzy

Conforme Araújo *et al.* (2007), os sistemas difusos estimam funções com descrição parcial do comportamento do sistema, onde especialistas podem prover o conhecimento heurístico, ou esse conhecimento pode ser inferido a partir de dados de entrada-saída do sistema. Pode-se dizer que os sistemas difusos são sistemas baseados em regras que utilizam variáveis lingüísticas difusas (conjuntos difusos) para executar um processo de tomada de decisão.

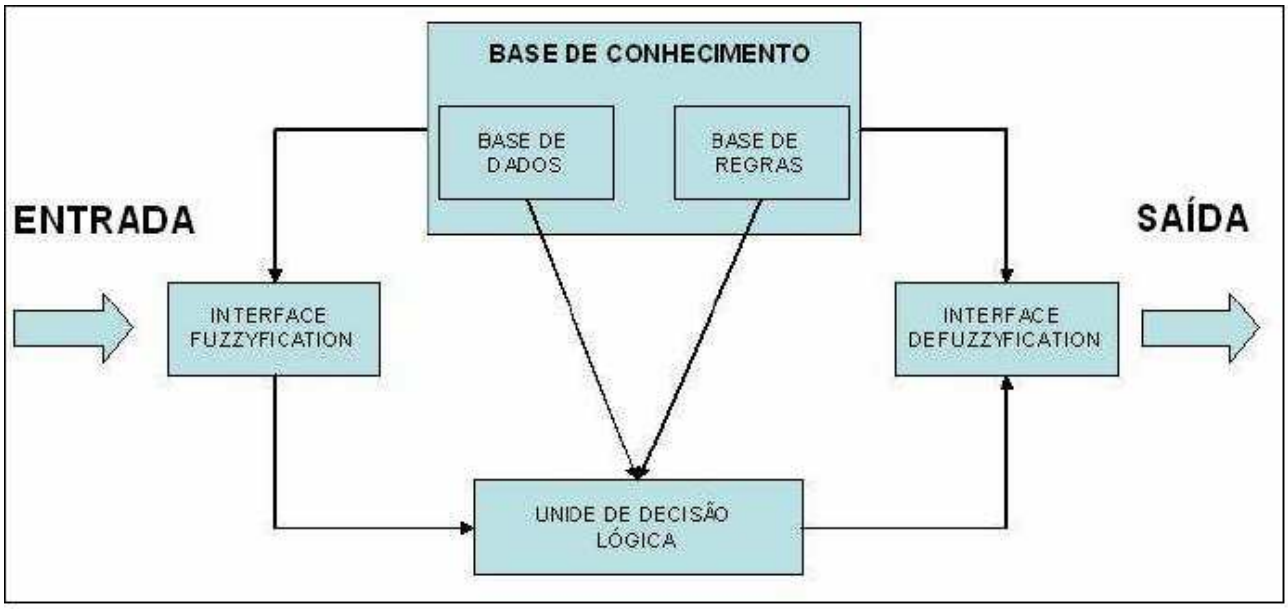


Figura 2. Um sistema de Inferência pode ser composto de 5 blocos principais
Fonte: Adaptado de Jang, (1993 *apud* ARAUJO *et al*, 2007).

A Figura 2 ilustra como são os blocos que compõem um sistema *fuzzy* descritos a seguir segundo (Araújo *et al,* 2007).

Base de Regras: contém um conjunto de regras/proposições *fuzzy* onde as variáveis antecedentes/conseqüentes são variáveis lingüísticas e os possíveis valores de uma variável lingüística são representados por conjuntos difusos. Uma proposição difusa expressa relações entre variáveis lingüísticas e conjuntos difusos, de acordo com Oliveira, (1999 *apud* ARAÚJO *et al*, 2007). As proposições podem estar no formato Se - Então, como por exemplo:

Se pressão é alta Então volume é pequeno

Onde pressão e volume são variáveis lingüísticas, alta é o antecedente (premissa) e volume é conseqüente.

Base de Dados: Define as funções de pertinência do conjunto difuso nas regras *fuzzy*.

Unidade de Decisão Lógica: realiza operações de inferência, para obter, a partir da avaliação dos níveis de compatibilidade das entradas com as condições impostas pela base de regras, uma ação a ser realizada pelo sistema.

Interface de Fuzzyficação (*Fuzzyfication Interface*): utilizando as funções de pertinência pré-estabelecidas, mapeia cada variável de entrada do sistema em graus de pertinência de algum conjunto difuso que representa a variável em questão.

Interface de Defuzzyficação (*Defuzzyfication Interface*): transforma os resultados difusos da inferência em valores de saída. Calcula a saída com base na inferência obtida no módulo Unidade de Decisão Lógica, com as funções de pertinência das variáveis lingüísticas da parte conseqüente das regras para obter uma saída não difusa. Nessa etapa as regiões resultantes são convertidas em valores de saída do sistema.

Implicação Lógica e Regras de Inferência

Segundo Shaw e Simões (1999), há um aspecto relevante na forma de pensar dos seres humanos: a implicação lógica, que consiste na formulação de uma conexão entre causa e efeito, ou uma condição e sua conseqüência. Em carreiras técnicas tais implicações lógicas são virtualmente encontradas em todas as situações, por exemplo, ao operar uma máquina, ao resolver problemas matemáticos, programar um computador, seguir um procedimento em um manual de instruções, etc. Nestes casos, segue-se consciente ou inconscientemente certas regras de inferência, da seguinte forma:

"SE o trânsito esta PESADO na Rua nº2 ENTAO mantenha o semáforo Verde MAIS TEMPO ACESO" onde o termo PESADO e MAIS TEMPO ACESO representam, conjuntos *fuzzy*. PESADO é uma função que define o grau de densidade do transito, enquanto MAIS TEMPO ACESO é uma função que define o grau de duração do tempo de operação do semáforo. O fato de se implantar "inteligência" no controlador do semáforo consiste então em associar termos *fuzzy* através de uma inferência *fuzzy*, expressa por uma estrutura SE... ENTÃO.

Fuzzyficação e Defuzzyficação Natural

Shaw e Simões (1999), afirmam que a operação das atividades humanas requer uma aproximação de dados e informações sensoriais através de termos vagos ou imprecisos. O cérebro humano codifica tais imprecisões naturais através de conjuntos e números *fuzzy*. Uma pessoa que esteja na função de operador de processos não precisa de um valor exato e definido para uma variável, por exemplo, a velocidade. Tal operador consegue classificar a informação em conjuntos, tipo BAIXA, MEDIA, e ALTA. Esses conjuntos representam valores "fuzzyficados" dos valores exatos da velocidade. A seguir o operador formula e executa uma estratégia de controle baseado na

compreensão de cada variável de entrada e de saída, o fluxo de dados no cérebro fica reduzido apenas ao que é necessário para se executar a tarefa requerida com a precisão e resolução necessária, assim o operador humano processa as quantidades *fuzzy*, chegando a uma variável *fuzzy* com sua ação de controle.

O ser humano naturalmente trabalha com características incertas, mas as máquinas, equipamentos, e controles industriais precisam de um número real que represente o valor de referencia necessário. Dessa maneira é necessário um processo de conversão do valor *fuzzy* resultante da saída da inferência – para um numero real, tal como acionar uma alavanca ou ajustar um botão para uma determinada posição, esse processo é chamado de defuzzyficação. Há vários exemplos que mostram que o ser humano tem essa capacidade de defuzzyficação natural, na habilidade de decifrar caligrafia, entender o linguajar com sotaque, ou até reconhecer pessoas após longo período de ausência (SHAW & SIMÕES, 1999).

Segundo Shaw e Simões (1999) está demonstrada que a habilidade de manipular conjuntos e números *fuzzy* é uma das atividades mais importantes desempenhadas pelo cérebro humano. Uma vez que não há modelo matemático para se seguir, o processo de decisão *fuzzy* tem um processo computacional mínimo. Provavelmente, essa habilidade desenvolveu-se através de gerações, já que o fato de se trocar precisão por velocidade é decisivo para sobrevivência biológica em situações críticas, ou perigos naturais.

Além dos fatos descritos, há uma característica que em sistemas complexos a precisão matemática perde seu significado. Em “Princípio da Incompatibilidade”, Zadeh (1978), afirma que “Conforme a complexidade de um sistema aumenta, nossa habilidade de fazer afirmações precisas e significativas sobre seu comportamento diminui, até um limiar em que a precisão e relevância tornam-se características mutuamente exclusivas”.

Variáveis Lingüísticas

Segundo Brandão (2005), a lógica *fuzzy* utiliza variáveis lingüísticas para mapear um determinado conjunto dentro de um universo. O raciocínio da lógica *fuzzy* é pensar de forma difusa, ou seja, criar regras numa linguagem natural, onde essas variáveis difusas recebam valores lingüísticos (ALTO, BAIXO, MÉDIO) que são transformadas em dados numéricos para a melhor interpretação do computador.

Uma variável lingüística está relacionada a um conjunto de dados numéricos *fuzzy* ou a uma função de pertinência, relacionando os termos lingüísticos aos graus de pertinência e assim possibilitando um significado numérico.

Com a utilização dessas variáveis é possível construir regras difusas do tipo: “Se a temperatura está alta e aumentando lentamente então diminua a temperatura do ar-condicionado um pouco”, por exemplo, para a recuperação de informações, tem-se: “Selecione todos os alunos que possuem nota alta”, note que, essas regras não possuem nenhum tipo de dado numérico tudo é escrito em linguagem natural.

Na Figura 3 pode-se visualizar uma representação *fuzzy* para a temperatura ambiente. Por exemplo, 18ºC não é totalmente frio nem totalmente agradável pode se dizer que 18ºC possui uma pertinência (μ) de 0,45 frio, e ao mesmo tempo 0,30 agradável. Assim 18ºC está mais para frio do que para agradável, mas não deixa de ser um pouco agradável Weber (2003 *apud* BRANDÃO, 2005).

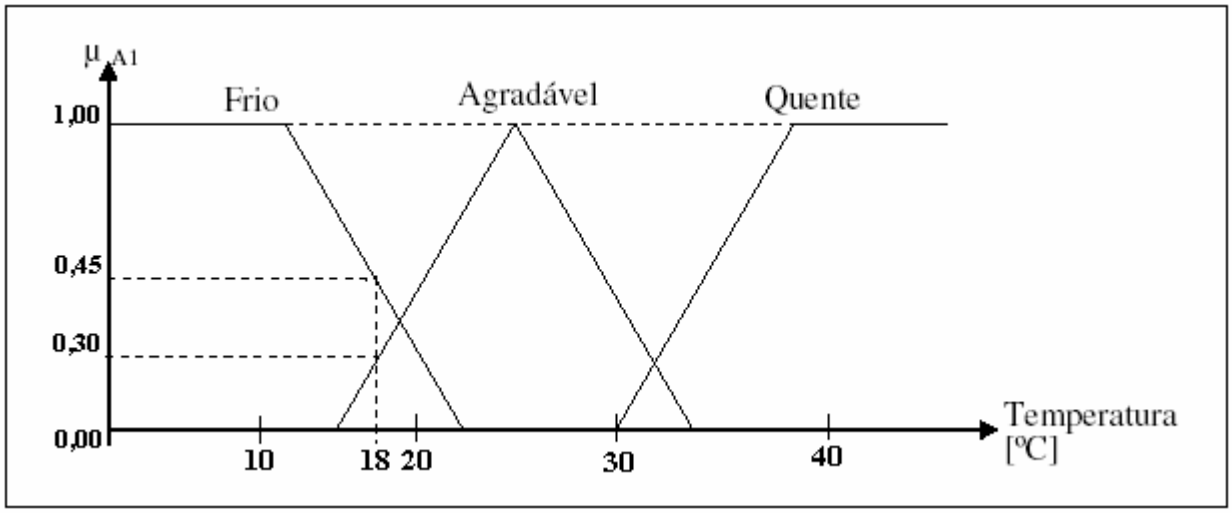


Figura 3. Gráfico fuzzy que representa a temperatura ambiente
Fonte: Adaptado de Weber (2003 *apud* BRANDÃO, 2005)

Num determinado conjunto universo de discurso, por exemplo: TEMPERATURA, pode-se ter n variáveis lingüísticas. No caso da Figura 3 pode ser acrescentando mais variáveis, por exemplo: MUITO FRIO e MUITO QUENTE.

2.1.2.4 Fundamentos Conjuntos Fuzzy

Conforme Ortega (2001), na teoria clássica dos conjuntos, o conceito de pertinência de um elemento a um conjunto fica bem definido. Dado um conjunto A em um universo X, os elementos deste universo simplesmente pertencem ou não pertencem àquele conjunto. Isto pode ser expresso pela função característica f_A, apresentado na Equação 1.

$$f_A(x) = \begin{cases} 1 \text{ se e somente se } x \in A \\ 0 \text{ se e somente se } x \notin A \end{cases} \qquad \text{Equação 1}$$

Zadeh (1978), propôs uma caracterização mais ampla, generalizando a função característica de modo que ela pudesse assumir um número infinito de valores no intervalo [0,1]. Um conjunto *fuzzy A* em um universo *X* é definido por uma função de pertinência $\mu_A(x)$: $X \rightarrow [0,1]$, e representado por um conjunto de pares ordenados (Equação 2).

$$A = \{\boldsymbol{\mu}_A(x)/x\} \qquad x \in X \qquad \text{Equação 2}$$

Onde $\mu_A(x)$ indica o quanto *x* é compatível com o conjunto *A*. Um determinado elemento pode pertencer a mais de um conjunto *fuzzy*, com diferentes graus de pertinência.

O *conjunto suporte* de um conjunto *fuzzy A* é o conjunto de elementos no universo *X* para os quais $\boldsymbol{\mu}_A(x)>0$. Um conjunto *fuzzy* cujo suporte é um único ponto *x*' com $\mu_A(x') = 1$ é chamado de conjunto unitário *fuzzy* ou *singleton*. Assim, um conjunto *fuzzy* também pode ser visto como o mapeamento do conjunto suporte no intervalo [0,1], o que implica em expressar o conjunto *fuzzy* por sua função de pertinência.

Conjuntos *fuzzy* podem ser definidos em universos contínuos ou discretos. Se o universo X for discreto e finito, o conjunto *fuzzy A* é normalmente representado:

- Por um vetor contendo os graus de pertinência no conjunto A dos elementos correspondentes de X ;
- Por meio da seguinte notação (que não deve ser confundida com a soma algébrica) (Equação 3):

Equação 3

$$\sum_{i=1}^{n} \mu_A(x_i) / x_i$$

Se o universo X for contínuo, emprega-se muitas vezes a seguinte notação (onde o símbolo de integral deve ser interpretado da mesma forma que o da soma no caso de um universo discreto) (Equação 4):

Equação 4

$$\int_X \mu_A(x) / x$$

Exemplos de Utilização da Lógica Fuzzy

Segundo Araújo *et al* (2007), pioneiros na utilização da lógica *fuzzy*, os japoneses já adquiriram grande conhecimento de sua utilização e a faz com freqüência, aperfeiçoando os aparelho mais simples como os aspiradores de pó, que agora usam controladores de 4 bits rodando algoritmos *fuzzy* que interrogam sensores de pó e ajustam o poder de sucção, e inventando novos e complexos mecanismos como câmeras fotográficas inteligente, que usa um dispositivo chamado *charge-coupled* (CCD) para medir a claridade das imagens em seis regiões do campo de visão e usa estas informações para determinar se a imagem está no foco ou não; ela também rastreia a taxa de mudança no movimento da lente durante a procura do foco e controla a velocidade para evitar *overshoot*. Este sistema possui 12 entradas: 6 para obter a claridade atual fornecendo dados para o CCD e 6 para medir a taxa de mudança no movimento da lente. A saída é a posição do curso da lente.

Outros exemplos que Araújo *et al* (2007) cita, são máquinas de lavar Hitachi que usam controladores *fuzzy* para controle de peso, verificação de tipo de tecido, e detecção de sujeira, que automaticamente adapta os ciclos de lavagem para o uso otimizado de potência, água, e detergente.

O sistema *fuzzy* não é utilizado apenas para facilitar nosso dia-a-dia, mas também para diminuirmos gastos nas execuções de tarefas, como é o exemplo do ar condicionado industrial

projetado pela Mitsubishi usa 25 regras de aquecimento e 25 regras de refrescamento. Comparado com outros tipos de ar condicionado, o controlador *fuzzy* aquece e esfria 5 vezes mais rápido, reduz o consumo de potência em 24%, sem contar que mantém a temperatura 2 vezes mais estável e usa menos sensores (ARAÚJO *et al* 2007).

A implementação da lógica *fuzzy* vem crescendo cada dia mais e nas mais diversas áreas, como no reconhecimento de caracteres, em sistemas óticos, robôs, helicópteros comandados por voz através de um robô, sistemas de elevadores, aterrissagem de naves espaciais, etc (ARAÚJO *et al* 2007).

3 DESENVOLVIMENTO

Neste capítulo serão descritos os processos para desenvolver o projeto, desde o seu planejamento, com levantamento de requisitos, análise de riscos até a parte de implementação como, padrões utilizados, linguagem, tecnologia para implementação.

O objetivo principal do sistema especialista com lógica *fuzzy* para o auxílio na manutenção de microcomputadores é, que a partir de um micro não-funcional, analisar seus componentes de hardware baseado em uma seqüência de passos, a fim de identificar e emitir um parecer a respeito da possível causa do problema baseado nos sintomas informados. A partir de uma pesquisa na literatura (TORRES, 1999) e de uma série de entrevistas com técnicos especialistas em manutenção de microcomputadores, foi obtida uma lista dos componentes (Apêndice 1. Tabela 9.) que apresentam mais problemas.

O sistema por sua vez orienta a substituição do componente, uma vez que para o conserto seria necessário conhecimento em eletrônica, não cabendo ao técnico em manutenção este conhecimento, o sistema também não armazena os resultados obtidos para posterior consulta.

Assim, a primeira etapa de modelagem foi a definição dos componentes que compõe o hardware de um microcomputador que o sistema irá avaliar. O hardware básico de um micro-computador é composto essencialmente de um monitor de vídeo, teclado, mouse e de um gabinete onde internamente estão instalados: fonte, placa mãe, processador, memória RAM, placa de vídeo, disco rígido (HD), *driver* de disquete e CD-ROM (TORRES, 1999).

3.1 Modelagem

Este capítulo está divido em três tópicos. Inicialmente, serão apresentados dados referentes a modelagem do sistema, como os requisitos funcionais e não-funcionais que foram levantados para o sistema, regras de negócio, os problemas abordados e sua soluções. Por fim, a análise de risco do projeto.

3.1.1 Metodologia

Durante a descrição deste projeto, realizou-se uma observação crítica do que seria necessário para o desenvolvimento do sistema, assim como suas limitações, descrito na seqüência.

3.1.1.1 Requisitos Funcionais

Requisitos funcionais compreendem as funcionalidades que os usuários querem ou precisam que o sistema ofereça, os seguintes itens foram considerados essenciais:

- O sistema propõe um determinado procedimento ou substituição do componente;
- O sistema orienta uma seqüência de testes a fim de diagnosticar a origem do problema; e
- O usuário informar os sintomas do problema do microcomputador para a posterior análise do sistema.

3.1.1.2 Requisitos Não-Funcionais

Os requisitos não-funcionais compreendem propriedades do sistema, como manutenibilidade, desempenho, custos entre outros. Como requisitos não-funcionais foram definidos os seguintes:

- O sistema deve operar em ambiente Windows 9x ou superior; e
- O sistema suporta performance estática, ou seja, apenas um usuário por vez.

3.1.1.3 Regras de Negócios

Com base nos sintomas informados o sistema deverá propor um determinado procedimento ou substituição do componente, o técnico com o auxílio do sistema, irá realizar o teste de cada componente a fim de diagnosticar onde está o problema.

O sistema faz referência aos seguintes hardwares e componentes: Fonte de alimentação, Monitor, Disco rígido e *Drivers* (CD-ROM e disquete), Placas *off-boards*, Placa de Vídeo, Memória, Placa-Mãe, Processador e por fim Teclado e Mouse. O bom funcionamento dos componentes e a boa interação entre eles são importantes para o bom funcionamento do micro-computador. A seguir é comentado um pouco mais sobre cada um deles assim como os problemas mais comuns encontrados (Torres, 1999).

Segundo Torres (1999), A fonte de alimentação do computador é projetada para transformar as tensões comuns da rede elétrica em níveis compatíveis da CPU, além de filtrar ruídos e

estabilizar os valores de tensão. A fonte de alimentação do microcomputador converte a tensão alternada AC 220/110 nas tensões contínuas (DC ou VDC) que alimentam as diversas placas e periféricos do computador. Os defeitos mais comuns são o fusível queimando e o ventilador que por vezes gera ruídos ou não gira corretamente, muitas vezes por acumulo de poeira.

Monitores, dispositivo de saída de dados padrão para visualização de dados pelo usuário, em geral não costumam apresentar problemas em sua parte lógica, em sua maioria os problemas estão relacionados a conexão com a rede elétrica ou com o computador, mau contatos são comuns nos cabos de forca, na parte de configuração resoluções não compatíveis e freqüências não suportadas também são comuns.

Discos rígidos, CD-ROM e disquetes, são dispositivos de armazenamento de dados de longo prazo, os problemas mais comuns encontrados nestes *drivers*, são a queima do dispositivo, ou problemas relacionados a mau contato no cabo de alimentação, assim como problemas com o *flat-cable* que os conecta a placa mãe.

Placas *off-board*, são placas que não possuem seus circuitos integrados a placa-mãe, sua conexão é feita através de *slot*, as mais usadas são placas de rede, fax modem e placa de som, os problemas mais comuns relacionados a estes componentes são o conflito com algum outro dispositivo já presente na placa mãe ou algum mau contato entre o *slot* e o dispositivo.

Placa de vídeo, instalada na placa-mãe, transmite os sinais do computador para o monitor, geralmente não apresenta defeito, porém nos modelos *off-board* algum mau contato pode ocasionar em um mau funcionamento ou até mesmo no não funcionamento da mesma.

Memória RAM, responsáveis pelo armazenamento de dados de curto prazo, por ser um componente que esta conectado na placa mãe por meio de um *slot* também apresenta mau contato, porém muitas vezes a memória apresenta mau funcionamento, ocasionando travamento do sistema, fazendo com que o usuário precise reiniciar o sistema operacional.

Placa mãe, considerado o elemento mais importante de um computador, pois tem como função permitir que o processador se comunique com todos os periféricos instalados, os problemas mais comuns encontrados são a presença de sujeira nos *slots* de conexão e anormalidades nos componentes, provenientes de oscilações bruscas na fonte de alimentação.

Teclado e *mouse*, dispositivos mais comuns de entrada de dados, responsáveis pela interação homem máquina, por serem dispositivos manuseados fisicamente costumam apresentar problemas com relativa freqüência, porém de fácil substituição e diagnóstico.

Processador, responsável pela manipulação e interpretação dos dados, costumam apresentar defeitos muito raramente, geralmente relacionado a super-aquecimento ou configuração fora das especificação do fabricante.

3.1.1.4 Fluxograma da Seqüência de Testes

A seqüência de testes representada pelo fluxograma (Figura 4) é similar a mesma seqüência utilizada pelos técnicos, onde o sistema leva em conta o grau de pertinência a cada pergunta, para ao final propor uma solução para o problema. Cada componente é fundamental para o funcionamento do microcomputador, por exemplo, se for identificado um problema na fonte de alimentação, nenhum outro dispositivo, com exceção do monitor, receberá a alimentação necessária para o seu correto funcionamento, logo será sugerida a troca do componente por um funcional, para que possa recomeçar o processo de análise.

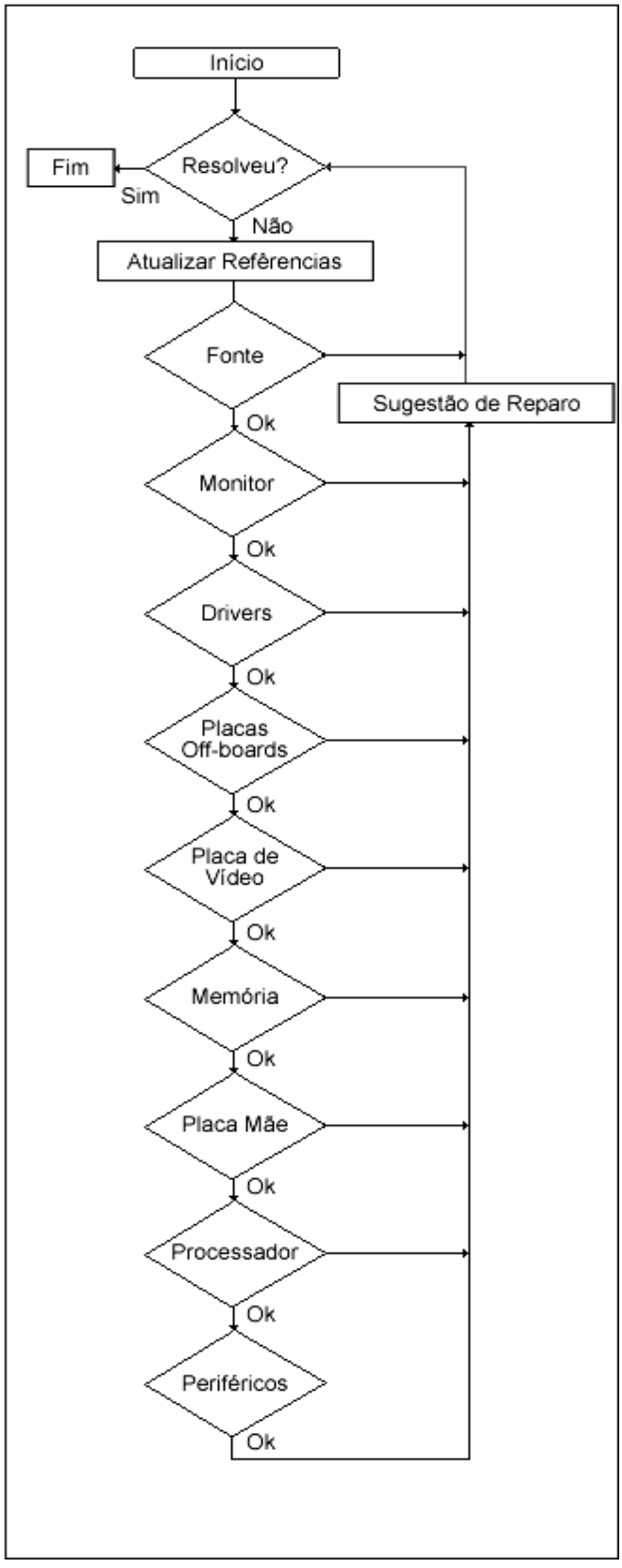


Figura 4. Fluxograma de Testes

3.1.1.5 Problemas Tratados pelo Sistema

Com base na literatura e em entrevistas onde foram abordados os casos mais comuns encontrados em seus ambientes de trabalho, especialistas que trabalham no setor de Tecnologia da Informação da Sociedade Educacional de Santa Catarina - SOCIESC levantaram os seguintes problemas:

- Computador não liga;
- Computador liga, mas não aparece nada no monitor;
- Computador liga normalmente, porém emite ruídos, como “rangidos” e “cliques”;
- Computador liga normalmente, porém CD-ROM não funciona;
- Computador liga normalmente, porém disquete não funciona;
- Identifica os componentes, porém não carrega o sistema operacional;
- Ao liga o computador, o mesmo emite “bips”, e nada aparece na tela;
- Não identifica as placas *off-boards* no sistema operacional;
- Não identifica algum dos *drivers*, disco rígido, cd-rom ou disquete;
- Exibe mensagem na tela de teclado não encontrado, ou teclado não responde;
- *Mouse* não funciona no sistema operacional;
- Computador ao ligar, fica reiniciando;
- Computador ao ligar, após algum tempo reinicia;
- Computador ao ligar, o monitor apresenta padrões de caracteres não reconhecíveis;
- *Drive* de disquete apresenta seu “*led*” ligado direto; e
- Computador esta ligado a um cabo de rede operante, porém ao entrar no sistema operacional a rede do computador não funciona e a configuração esta OK no Sistema Operacional.

3.1.1.6 Sugestões de reparo

Para cada componente identificado com mau funcionamento, existe um procedimento a ser adotado a fim de sanar o problema. Nas entrevistas, especialistas também descreveram qual procedimento mais indicado para cada problema.

Fonte

Os problemas mais comuns envolvendo este componente são a queima do fusível e a obstrução do ventilador responsável pelo seu resfriamento.

Como a grande maioria das fontes de alimentação contém o fusível na parte interna da fonte e o mesmo se encontra integrado ao circuito lógico da fonte. Sua troca deve ser feita por profissionais especializados, contudo nas fontes que possuem o fusível na parte externa, é possível a sua simples substituição.

Para problemas referentes ao superaquecimento, o ventilador deve estar com acúmulo de sujeira ou não funcionando corretamente. No caso de sujeira uma simples limpeza com um pincel pode resolver o problema, porém se o ventilador não estiver funcionando, a fonte deve ser substituída.

Por fim uma checagem nas voltagens de saída da fonte com auxílio de um multímetro pode exibir um mau funcionamento de algum componente interno da fonte, fazendo necessária a sua substituição.

Monitor

Monitores de forma geral, não costumam apresentar problemas em seus componentes, seus problemas geralmente estão relacionados a configurações que o dispositivo não é compatível (como resoluções muito altas, e freqüências muito alta) e aos cabos de sua conexão. O mau contato no cabo de alimentação pode ser resolvido facilmente com um ajuste no cabo ou sua simples substituição se o problema permanecer. Problemas com resolução ou freqüência fora dos padrões do fabricante do monitor, pode ser alterada na configuração da placa de vídeo. Por fim a verificação do correto funcionamento do botão de liga/desliga.

Drivers

Dispositivos como o disco rígido possuem uma vida útil bastante prolongada, porém com uma oscilação brusca no fornecimento de energia, pode vir a queimar, obrigando a sua substituição.

Outro problema comum é o mau contato no cabo de alimentação do dispositivo, que com o manuseio pode vir a não encaixar perfeitamente, podendo ser resolvido com um ajuste ou substituição do cabo se possível. O *flat-cable,* por onde passam os dados do disco, também pode apresentar problemas, necessitando a sua substituição.

Drivers de cd-rom e disquete, como o disco rígido podem apresentar "mau contato" no cabo de alimentação e de transmissão de dados. *Drivers* de disquete em geral costumam ter uma vida útil bastante prolongada, por outro lado *drivers* de cd-rom costumam apresentar problemas de leitura com mais facilidade, ocasionando em uma troca do dispositivo.

Placas off-boards

Placas *off-boards* podem causar instabilidade no sistema devido a conflitos com outros dispositivos, neste caso verificar a configuração de cada dispositivo a fim de isolar os dispositivos conflitantes para a devida alteração na configuração. Por serem conectadas fisicamente na placa mãe por meio de *slots*, podem apresentar mau contato, algumas vezes a conexão em outro *slot* disponível resolve o problema.

Placa de vídeo

Quando *off-board*, pode apresentar problemas com mais facilidade, pois sua conexão por meio de *slots* na placa mãe possibilita o mau contato e/ou acúmulo de sujeira no mesmo, a limpeza ou troca de *slot* (se possível, pois muitas placas de vídeo são do padrão AGP ou PCIe, e os fabricantes de placa mãe disponibilizam apenas um *slot* destes padrões na placa). Nas placas *on-board* muitas vezes apenas a restauração das configurações de fábrica da placa mãe soluciona o mau funcionamento, se nenhum procedimento adotado sanar o problema a substituição do componente é inevitável.

Memória

Se os indícios indicarem problemas com a memória, a primeira coisa a ser feita é uma limpeza nos *slots* do "banco de memória" da placa mãe, assim como do pente de memória. Se o problema persistir é recomendável a troca do *slot* se possível, a fim de certificar-se que o problema não é no *slot* da placa mãe. Existem ainda softwares que testam os pentes de memória a procura de alguma imperfeição em seu funcionamento, e por fim a substituição da memória se for necessário.

Placa mãe

Em geral apresentam uma vida útil longa, apresentando problemas ocasionados por oscilações na fonte de alimentação e mau uso.

Após desconectar todos os componentes dispensáveis para o funcionamento do computador, ou seja deixando apenas placa de vídeo, processador, memória e a fonte de alimentação, testar um a um os componentes a fim de isolar o componente com problema. A restauração das configurações de fábrica. Muitas vezes resolvem problemas de configurações erradas ou problemáticas. A presença de alguma anomalia física dos componentes como capacitores fora do normal, como “estufados” ou queimados, também são indícios de mau funcionamento e requerem a substituição do componente.

Teclados e mouses

Por servem dispositivos de manuseio contínuo, apresentam problema com relativa freqüência, o dispositivo pode estar mau encaixado na placa mãe do computador, isso faria com que tornassem inoperantes de uma hora para outra. No caso de *mouses* ópticos, a limpeza do sistema deslizante não é necessária, contudo nos modelos mais antigos (não ópticos), uma simples limpeza em seu sistema de navegação pode resolver o problema precisão nos movimentos do cursor. Teclados com muito uso podem apresentar teclas presas ou teclas que não funcionam mais, necessitando a troca do teclado, uma limpeza periódica pode aumentar a vida útil do dispositivo.

Processador

Não costumam apresentar problemas, porém o principal sintoma de problema com processador é o da máquina ligar e não apresentar “bip” algum e não apresentar nada na tela. Com o excesso de calor costumam fazer com que o computador reinicie, necessitando de uma troca ou limpeza do dissipador de calor e do *cooler* do mesmo. Alguns modelos mais antigos de processador, que causam a “reinicialização” da máquina, funcionam melhor com a sua freqüência reduzida, muitas vezes resolvendo o problema, porém não é normal que o processador só funcione nestas condições, mostrando assim que a placa mãe pode estar comprometida.

3.1.1.7 Regras

Para a elaboração das regras foram utilizadas variáveis lingüísticas. Como definido anteriormente, os níveis de pertinência podem variar de acordo com o componente analisado, a fim de melhor atender as descrições dos sintomas apresentados.

As regras procuraram atender da melhor forma as possibilidades inerentes de cada componente analisado. Um exemplo de regra referente a um problema apresentado pelo CD-ROM do computador seria expresso como na Tabela 2.

Tabela 2. Exemplo de Regra.

Componente / Variável	**Não Lê**	**Erros**	**Lê**
CD-ROM	Ruim	Médio	Bom

Considerando que o usuário do sistema já tenha feito os testes necessários no CD-ROM, a fim de definir se seu funcionamento está de acordo com o esperado, é informado ao sistema como o componente se comporta ao ler uma mídia compatível qualquer.

Para melhor ilustrar, é utilizado no exemplo a seguir os testes necessários para diagnosticar se o componente “fonte de alimentação” esta funcionando corretamente, para isso serão feitos seis testes, onde serão realizadas algumas medições. Para melhor ilustrar as inferências realizadas assim como os conjuntos *fuzzy* foi utilizada a *Fuzzy Tollbox do MATLAB*©.

A Figura 5 representa o teste da saída 12v da fonte, onde uma variação de até 5% é considerada normal, contudo até uma variação de 10% a maioria das fontes ainda funcionam corretamente.

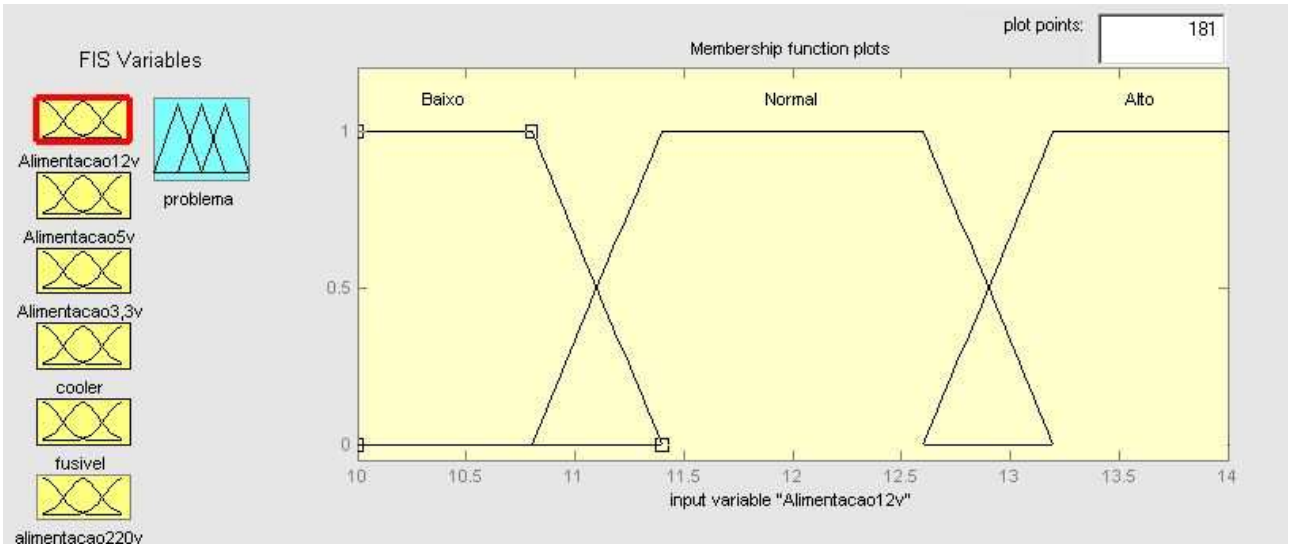


Figura 5. Teste da saída 12v

Na Figura 6 está representada o teste da saída 5v da fonte, onde também é normal uma variação de 5% e tolerável até 10%.

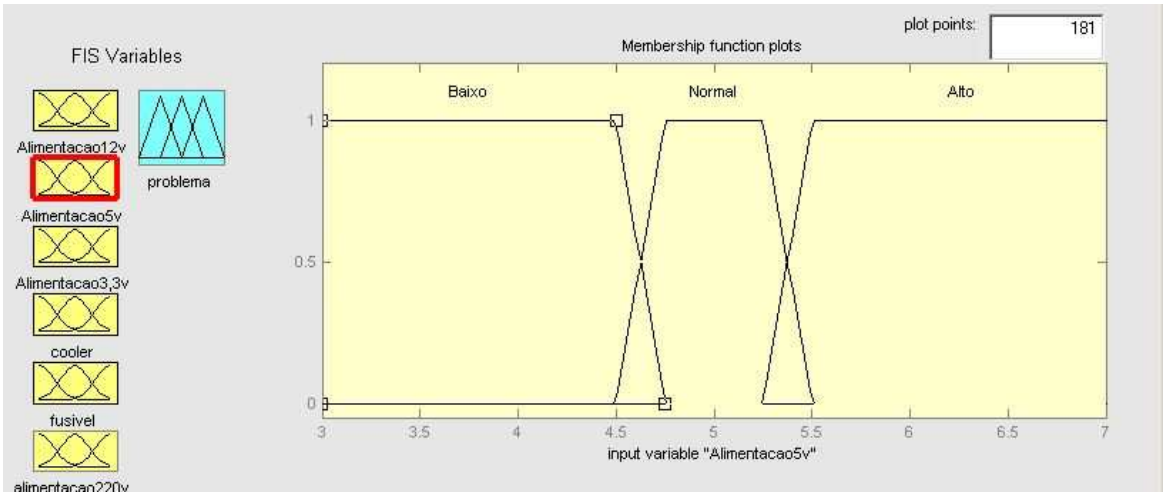


Figura 6. Teste da saída 5v

A Figura 7 representa o teste da saída 3.3v da fonte, onde uma variação de 5% a 10% é considerada normal para seu funcionamento.

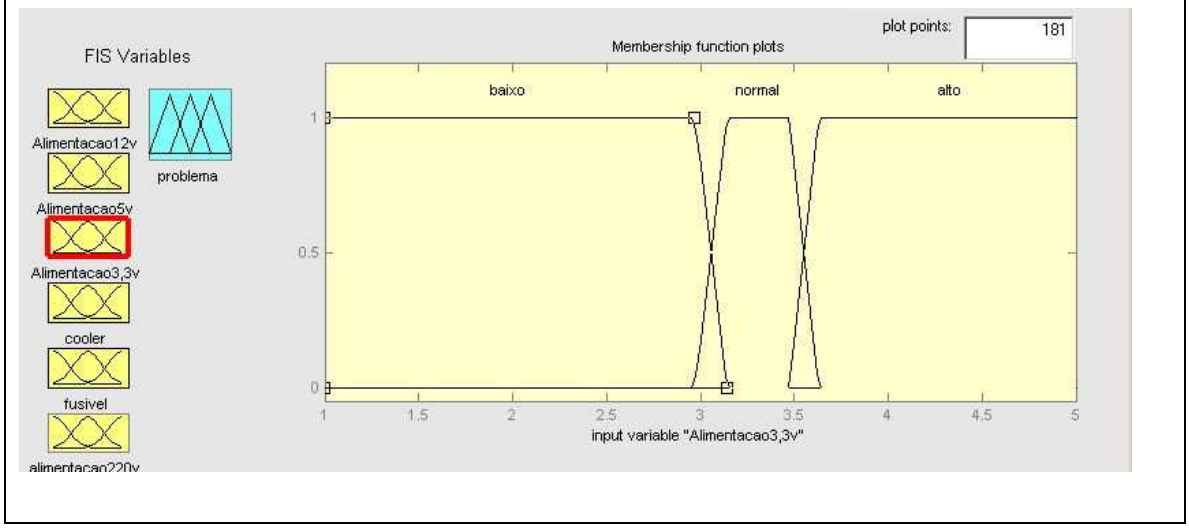


Figura 7. Teste da saída 3.3v

Para o teste do *cooler*, representado na Figura 8, foi levada em consideração a temperatura da fonte, pois com o aquecimento, a sua eficiência fica comprometida. A maioria das fontes existentes no mercado (fontes de baixo custo) opera entre 25~30ºC sendo que acima desta temperatura ocorre um progressivo decréscimo de potência até chegar a zero. Contudo abaixo dessa temperatura a fonte atinge sua máxima eficiência.

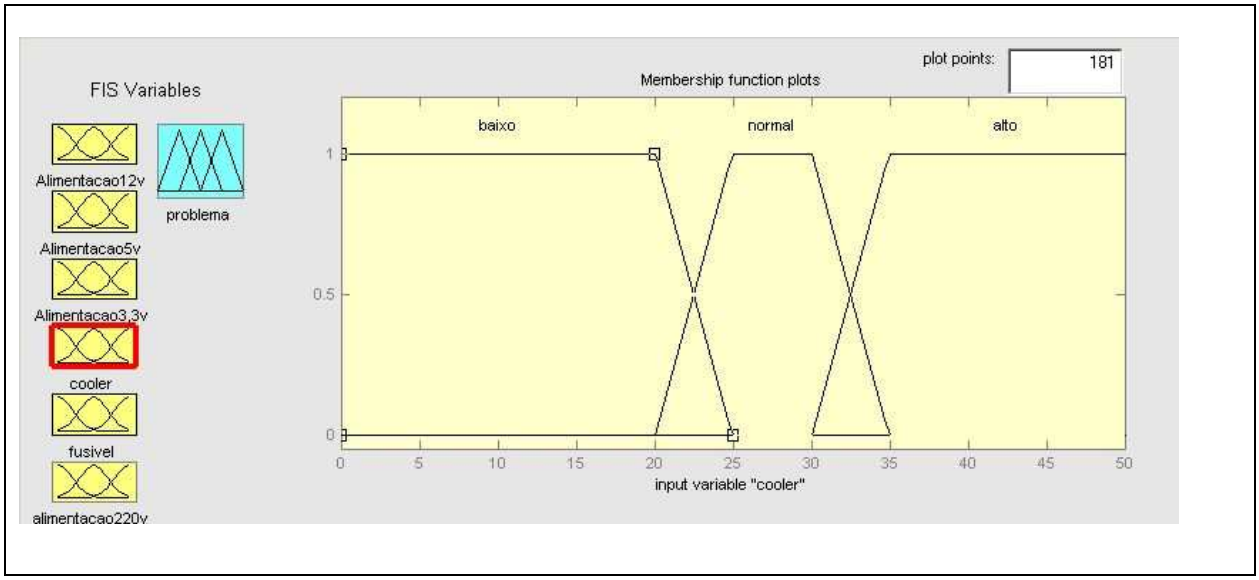


Figura 8. Teste do cooler

O teste do fusível pode ser feito através de uma regra determinística, ou seja, o fusível está ou não rompido. Porém, o fusível apresenta também uma incerteza em relação a corrente, que causa

o seu rompimento interrompendo a sua operação, por exemplo, 90% da corrente normal. O fusível da fonte do microcomputador é projetado para 3A, logo foi adotada a margem de erro de 10%, porém abaixo dessa corrente o fusível opera normalmente, como representada na Figura 9.

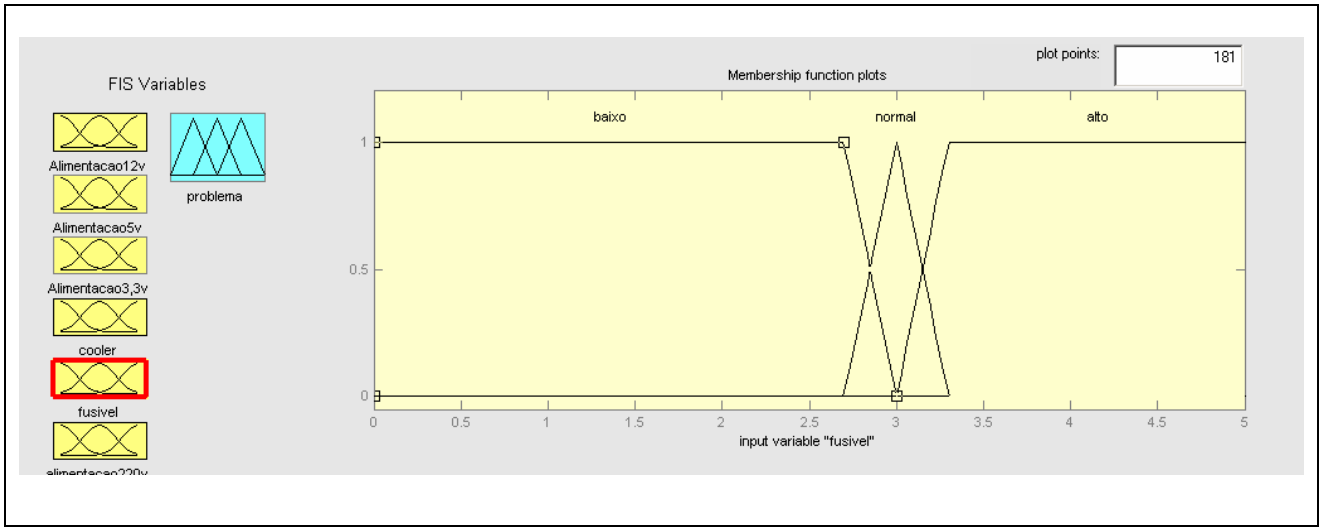


Figura 9. Teste do fusível

A Figura 10 representa a alimentação proveniente da rede. Para o estado de Santa Catarina, por meio de contrato a empresa fornecedora tem obrigação de fornecer energia entre 180~233V. Algo diferente desse intervalo a mesma possui um prazo de três meses para efetuar os devidos reparos. Contudo a tensão normal de funcionamento é entre 201 - 229V o que é admissível para o funcionamento da maioria dos equipamentos, dentre eles a fonte do computador. Estes valores são estabelecidos pela agência reguladora de energia elétrica ANEEL.

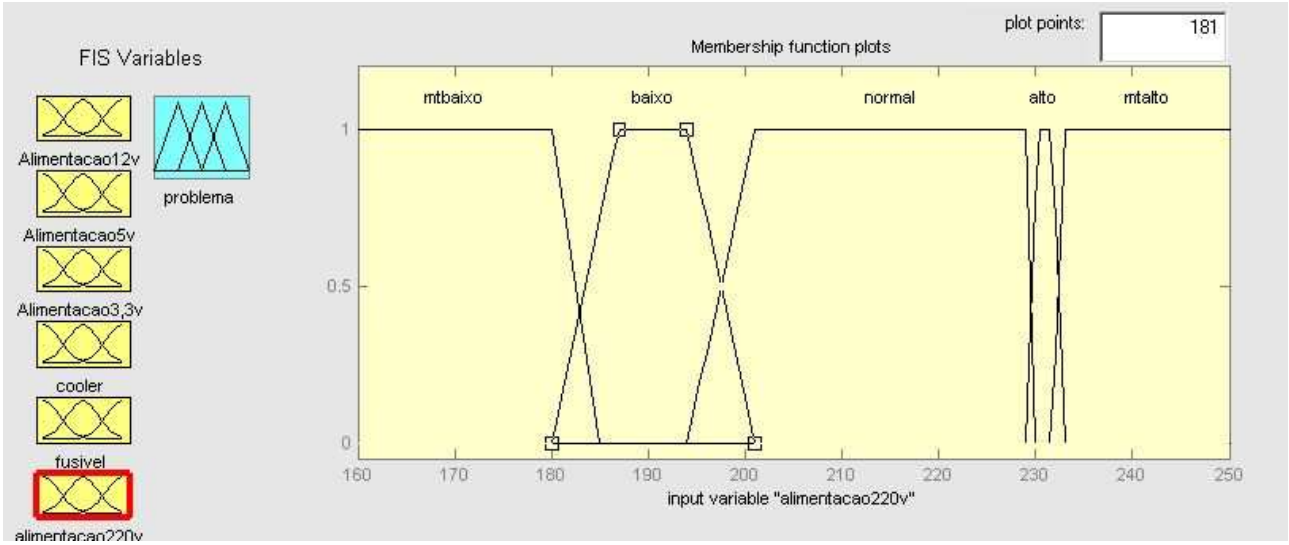


Figura 10. Teste da alimentação 220v AC

A Figura 11 demonstra como está o funcionamento da fonte. As três áreas triangulares representam os estados de operação da fonte, onde: BOM, representa funcionamento abaixo da carga projetada; NORMAL, representa funcionamento normal; e RUIM, representa que a fonte está operando fora das especificações para a qual foi projetada.

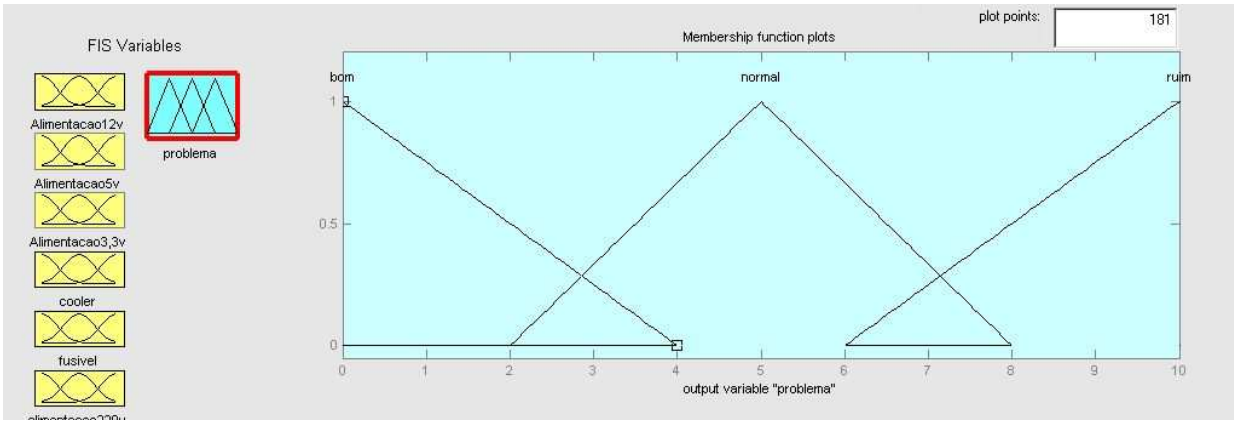


Figura 11. Saida do teste da Fonte.

Baseado nos testes descritos acima e com os valores de entrada, 12.48V, 4.89V, 3.29V, 30º, 2.5A, 212V; o sistema retorna o valor 6.47, informando que a fonte tem 64,7% de chance de estar

operando com problema ou seja, a fonte esta operando com normalmente, pois o valor encontrado possui um grau de pertinência maior no conjunto normal.

Para a simulação do problema da fonte, foram utilizados o método de Mamdani, na etapa de inferência, e o método do centro de gravidade (centróide), na etapa de defuzzyficação, pois serão os mesmos utilizados para a implementação do trabalho. A característica principal dos sistemas tipo Mamdani é que tanto o antecedente quanto o conseqüente são expressos por termos lingüísticos, transformando uma informação qualitativa em uma informação qualitativa através do conjunto *fuzzy* gerado durante o processo de inferência pode então ser utilizado diretamente em um diagnóstico. O método da centróide pode ser entendido como uma média ponderada onde $\Delta(x)$ funciona como peso do valor x, ou seja é calculado o centro de gravidade da área formada na inferência (SILVA & BASSANEZI, 2006).

A fim de identificar a uma maior quantidade de possíveis problemas o sistema foi implementado com a S*hell* FUZZYCLIPS, com aproximadamente 170 regras, dividida em 11 módulos, os quais representam os componentes verificados pelo sistema, como descrito anteriormente.

Para o teste do módulo fonte foram utilizadas 12 regras, como consta um exemplo da Figura 12, no módulo fonte para a avaliação é utilizado 6 fatos de entrada.

```
(defrule r-1-10-FONTE
        (declare (salience 800))
        (v1_ALIMENTACAO12V Normal)
        (v1_ALIMENTACAO5V Normal)
        (v1_ALIMENTACAO3.3V Normal)
        (v1_COOLER Baixo)
        (v1_FUSIVEL Normal)
        (v1_ALIMENTACAO220V Normal)
=>
        (assert (v1_FONTE Bom))
)
```

Figura 12. Exemplo de regra do módulo Fonte.

O teste do módulo monitor foram utilizados 13 regras, como consta um exemplo da Figura 13, para o módulo monitor são utilizados 3 fatos de entrada e as variáveis lingüísticas utilizadas são, bom, mau_contato e ruim.

```
(defrule r-1-4-MONITOR
        (declare (salience 756))
        (v1_ALIMENTACAO220VMO Normal)
        (v1_CONEXAORGB Normal)
        (v1_BOTAOPOWER Normal)
=>
        (assert (v1_MONITOR Bom)))
```

Figura 13. Exemplo de regra do módulo Monitor.

O módulo CD-ROM utilizou 26 regras para o seu teste, uma das regras pode ser vista na Figura 14, para o teste do módulo foram utilizados 6 fatos de entrada e as variáveis lingüísticas utilizadas foram, não_le, erros, bom, ruim, mau_contato, certo e errado.

```
(defrule r-1-3-CDROM
        (declare (salience 742))
        (v1_MIDIA Bom)
        (v1_BIOS Bom)
        (v1_CABOFLATCD Bom)
        (v1_CABOFORCACD Bom)
        (v1_CABOAUDIO Bom)
        (v1_JUMPERCD Certo)
=>
        (assert (v1_CDROM Bom))
)
```

Figura 14. Exemplo de regra do módulo CD-ROM.

No módulo HD foram utilizadas 18 regras e variáveis lingüísticas como, não_reconhece, reconhece_errado, reconhece_certo, ruim, mau_contato, bom, drive_errado, certo, errado, para o seu teste, uma das regras pode ser vista na Figura 15.

```
(defrule r-1-2-HD
        (declare (salience 732))
        (v1_BIOSHD ReconheceCerto)
        (v1_CABOFORCAHD Bom)
        (v1_CABOFLATHD Bom)
        (v1_JUMPERHD Certo)
=>
        (assert (v1_HD Bom))
)
```

Figura 15. Exemplo de regra do módulo HD.

O módulo FLOPPY é composto de 14 regras, um exemplo de uma delas pode ser vista na Figura 16, para o seu teste foi utilizado 3 fatos de entrada e variáveis lingüísticas como, ligado, desligado, bom, ruim e mau_contato.

```
(defrule r-1-3-FLOPPY
        (declare (salience 724))
        (v1_LED Desligado)
        (v1_CABOFLATFLOPPY Bom)
        (v1_CABOFORCAFLOPPY Bom)
=>
        (assert (v1_FLOPPY Bom))
)
```

Figura 16. Exemplo de regra do módulo FLOPPY.

No módulo Placas Off-boards foi utilizado 15 regras, um exemplo de uma delas pode ser vista na Figura 17, onde foram utilizados 3 fatos de entrada as variáveis lingüísticas utilizadas foram bom, médio e ruim.

```
(defrule r-1-3-PLACAOFF
        (declare (salience 716))
        (v1_LIMPEZAPLACA Bom)
        (v1_LIMPEZASLOT Bom)
        (v1_MAUCONTATO Bom)

=>
        (assert (v1_PLACAOFF Bom))
)
```

Figura 17. Exemplo de regra do módulo Placas Off-boards.

O módulo Placa de Vídeo é composto de 11 regras, o exemplo de uma delas pode ser visto na Figura 18, para o seu teste foi utilizado 3 fatos de entrada e variáveis lingüísticas como, funcionou, médio e não_funcionou.

```
(defrule r-1-3-PLACADEVIDEO
        (declare (salience 710))
        (v1_SLOTVIDEO Funcionou)
        (v1_TROCA Funcionou)
=>
        (assert (v1_PLACADEVIDEO Bom))
)
```

Figura 18. Exemplo de regra do módulo Placa de Vídeo.

O módulo memória é composto de 23 regras, o exemplo de uma delas pode ser visto na Figura 19, para o seu teste foi utilizado 5 fatos de entrada foram utilizadas variáveis lingüísticas como, ruim, médio, bom, sim, não e alguns.

```
(defrule r-1-3-MEMORIA
        (declare (salience 698))
        (v1_LIMPEZADOPENTE Bom)
        (v1_LIMPEZADOBANCO Bom)
        (v1_TROCADEBANCO Bom)
        (v1_TROCADAMEMORIA Bom)
        (v1_TESTEMEMORIA Bom)
=>
        (assert (v1_MEMORIA Bom))
)
```

Figura 19. Exemplo de regra do módulo Memória.

No módulo placa mãe foi utilizado 14 regras, o exemplo de uma delas pode ser visto na Figura 20, para o seu teste foi utilizado 3 fatos de entrada e variáveis lingüísticas como, sim, não, alguns e mau_contato.

```
(defrule r-1-3-PLACAMAE
        (declare (salience 690))
        (v1_BIOSPLACAMAE Sim)
        (v1_ANOMALIAS Bom)
        (v1_ALIMENTACAOMB Bom)
=>
        (assert (v1_PLACAMAE Bom))
)
```

Figura 20. Exemplo de regra do módulo Placa Mãe.

No módulo processador foi utilizado 11 regras, o exemplo de uma delas pode ser visto na Figura 21, para o seu teste foi utilizado 2 fatos de entrada foram utilizadas também variáveis lingüísticas como, baixa, normal, alta e overclock.

```
(defrule r-1-3-PROCESSADOR
        (declare (salience 684))
        (v1_TEMPERATURA Baixa)
        (v1_CLOCK Normal)
=>
        (assert (v1_PROCESSADOR Bom))
)
```

Figura 21. Exemplo de regra do módulo Processador.

O módulo periféricos foram utilizadas 11 regras, o exemplo de uma delas pode ser visto na Figura 22, para o seu teste foi utilizado 2 fatos de entrada e variáveis lingüísticas, bom, médio e ruim.

```
(defrule r-1-3-PERIFERICO
        (declare (salience 678))
        (v1_PORTA Bom)
        (v1_COMPONENTE Bom)
=>
        (assert (v1_PERIFERICO Bom))
)
```

Figura 22. Exemplo de regra do módulo Periférico.

3.1.1.8 FUZZYCLIPS

Para a implementação do sistema foi escolhido o FUZZYCLIPS, que consiste em uma extensão da S*hell* de sistema especialista CLIPS 6.02, baseado em lógica *fuzzy*, que foi desenvolvido pela *Integrated Reasoning Group* do *Institute for Information Technology* da *National Research Council of Canada*. O maior melhoramento feito com o FUZZYCLIPS foi um o mecanismo nebuloso para raciocínio que está totalmente integrado com o mecanismo de inferências e de fatos do CLIPS, permitindo assim representar e manipular fatos e regras nebulosos. FUZZYCLIPS pode trabalhar com raciocínio exatos, nebulosos (não-exato) e combinações destes, permitindo a mistura de termos nebulosos e normais em regras e fatos do sistema especialista. O FUZZYCLIPS possui interface é bastante simples como pode ser vista na Figura 23.

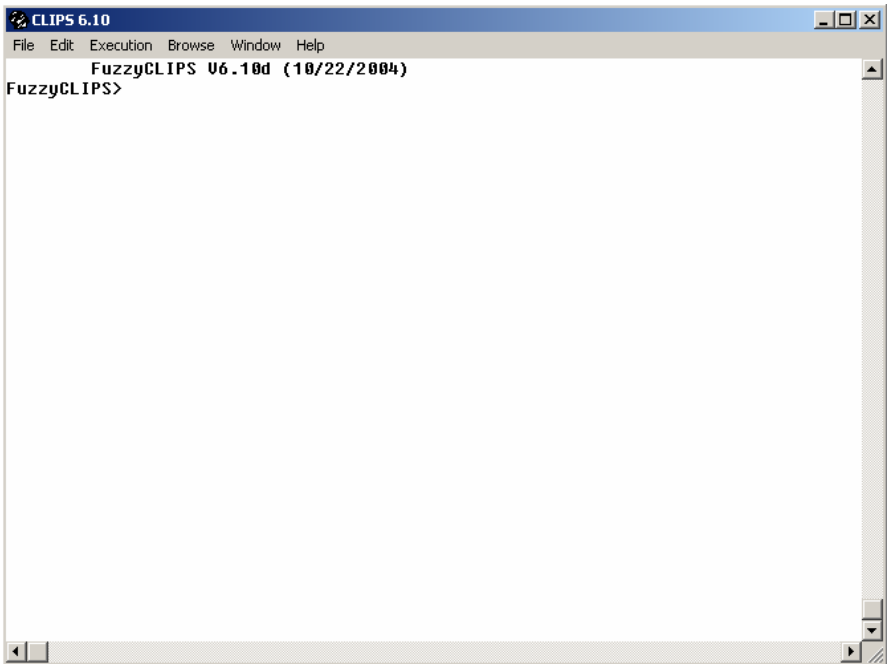


Figura 23. Tela principal do FUZZYCLIPS

Etapas para a implementação de um sistema especialista utilizando o FUZZYCLIPS

Segundo Fernandes (1995), existe algumas etapas importantes para a confecção do sistema especialista utilizando o FUZZYCLIPS:

1) Definição das variáveis globais

As variáveis que serão utilizadas por várias funções e módulos no sistema, bem como as variáveis manipuladas pelas funções de interface, devem ser definidas no início do programa através da construção *defglobal*, como consta no exemplo na Figura 24.

```
(defglobal
      ?*variavel1*= 0
      ?*variavel2*=" "
      ......)
```

Figura 24. Exemplo de definição de variáveis globais

2) Definição das funções criadas pelo usuário

Todas as funções criadas pelo usuário ou engenheiro do conhecimento para interface com usuário, cálculos, etc. São definidas através da construção *deffunction.* Logo após a definição das variáveis globais, como consta no exemplo na Figura 25.

```
(deffunction pergunta-simpbolo (?pergunta $?valores-permitidos)
(bind ?res (get-text-from-user ?pergunta))
(bind ?resposta nome)
(if (eq ?res “”)
       then (halt)
              else (bind ?resposta (string-to-symbol (lowcase ?res)))
)
?resposta)
```

Figura 25. Exemplo de funções criadas pelo usuário

3) Definição dos conjuntos difusos utilizados pelo sistema

Todas as variáveis lingüísticas utilizadas no sistema têm seus conjuntos difusos determinados através da construção *deftemplate*. Aqui também é definido o universo de discurso das variáveis lingüísticas, como consta no exemplo na Figura 26.

```
(deftemplate VL
       0 10 ud
(      (conjdifuso1 (0 0) (1 1) (2 0)
       (conjdifuso1 (1 0) (2 1) (3 0)
       (conjdifuso1 (2 0) (3 1) (4 0)
       (conjdifuso1 (3 0) (4 1) (5 0)
       ...) )
```

Figura 26. Exemplo de definições dos conjuntos difusos

4) Regra de startup

Este passo consiste em determinar a regra que iniciará a execução do programa. Esta regra é diferenciada das outras pela ausência de antecedentes. Dentro do conseqüente desta regra deve constar os módulos onde encontram as regras e as chamadas das funções a serem usadas durante a execução do sistema, como consta no exemplo na Figura 27.

```
(defrule startup
=>
       (load ”teste1.clp”)
       (pergunta-simbolo)
       ...)
```

Figura 27. Exemplo de regras de startup

5) Definição das regras

As regras a serem utilizadas pelo sistema devem ser definidas através da construção *defrule*. Estas regras podem ser definidas no corpo do programa principal (se forem em pequeno número), mas geralmente são definidas em módulos separados, os quais são carregados na regra *startup*, como consta no exemplo na Figura 28.

```
(defrule exemplo
       (VL conjdifuso1)
=>
       (printout t “Bem Vindo ao FuzzyCLIPS”))
```

Figura 28. Exemplo de definições de regra

Estas são as etapas básicas, que devem constar em todo programa em FUZZYCLIPS. Outras etapas podem ser necessárias, dependendo da necessidade da aplicação.

4 RESULTADOS

Baseado em dados coletados junto a Universidade do Vale do Itajaí – UNIVALI através do seu sistema de chamado foram realizados testes a fim de certificar as respostas dadas pelo sistema, uma vez que com base nos problemas relatados, o sistema teria que sugerir uma ação corretiva similar a efetuada pelo técnico humano.

O relatório possui 1732 chamados sobre problemas de hardware, coletados no período de 04/06/2007 a 04/06/2008 com a distribuição vista na Figura 29.

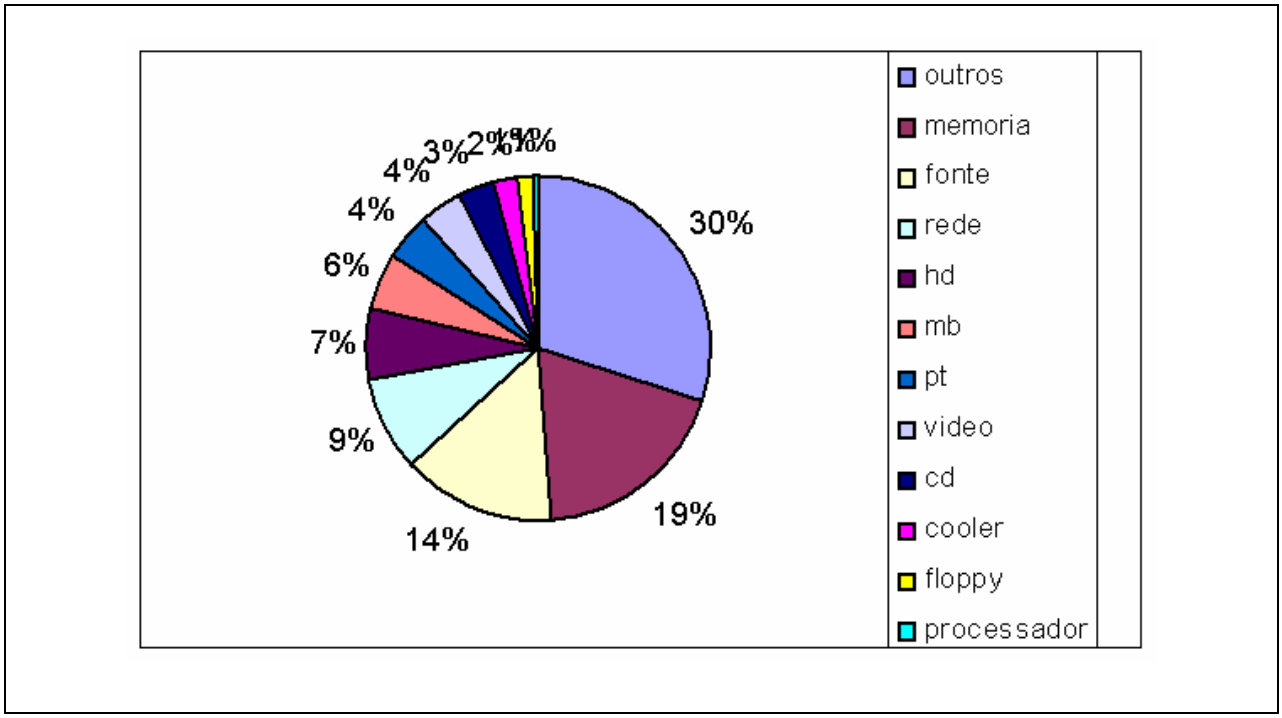


Figura 29. Distribuição de problemas de Hardware

A validação foi realizada por amostragem, onde foram obtidos 100% de acerto comparado com a ação corretiva do técnico humano, aqui apenas estão alguns exemplos dos casos e de como o sistema orienta o técnico no diagnóstico dos problemas.

4.1 Teste 01

Primeiro teste foi em relação a um computador que o usuário afirma não estar ligando. Ao chegar no local o técnico confirma a informação do usuário e efetua alguns testes a fim de verificar

o foco do problema, após alguns testes o técnico verifica que a fonte do computador deve ser substituída (Tabela 3).

Tabela 3. Chamado 01

Descrição do problema	Fato	Ação Corretiva
... “computador não está ligando” ...	Computador não liga.	A fonte estava queimada. Foi substituída.

Inicialmente, é necessário que o técnico efetue algumas medições com auxílio de um voltímetro, com o qual ele vai testar primeiramente a ausência de energia elétrica na tomada a qual o computador esta ligado, a tensão medida então é informada ao sistema através da entrada de fatos, posteriormente é medido a saída 12v, 5v e 3,3v responsáveis pela alimentação dos periféricos e placa-mãe, por ultimo é inserido o fato relativo a temperatura da fonte e a corrente que esta passando pela fonte.

Ao executar o sistema o mesmo analisa os fatos inseridos na base de fatos, que após avaliar as regras r-1-1-FONTE a r-5-10-FONTE o sistema verifica que a regra que melhor se encaixa nos fatos informados é a r-5-10-FONTE, exibindo a mensagem "O componente fonte esta ruim, verifique alimentação, cabos e temperatura. Se o problema persistir, substitua o componente", ilustrada pela Figura 30.

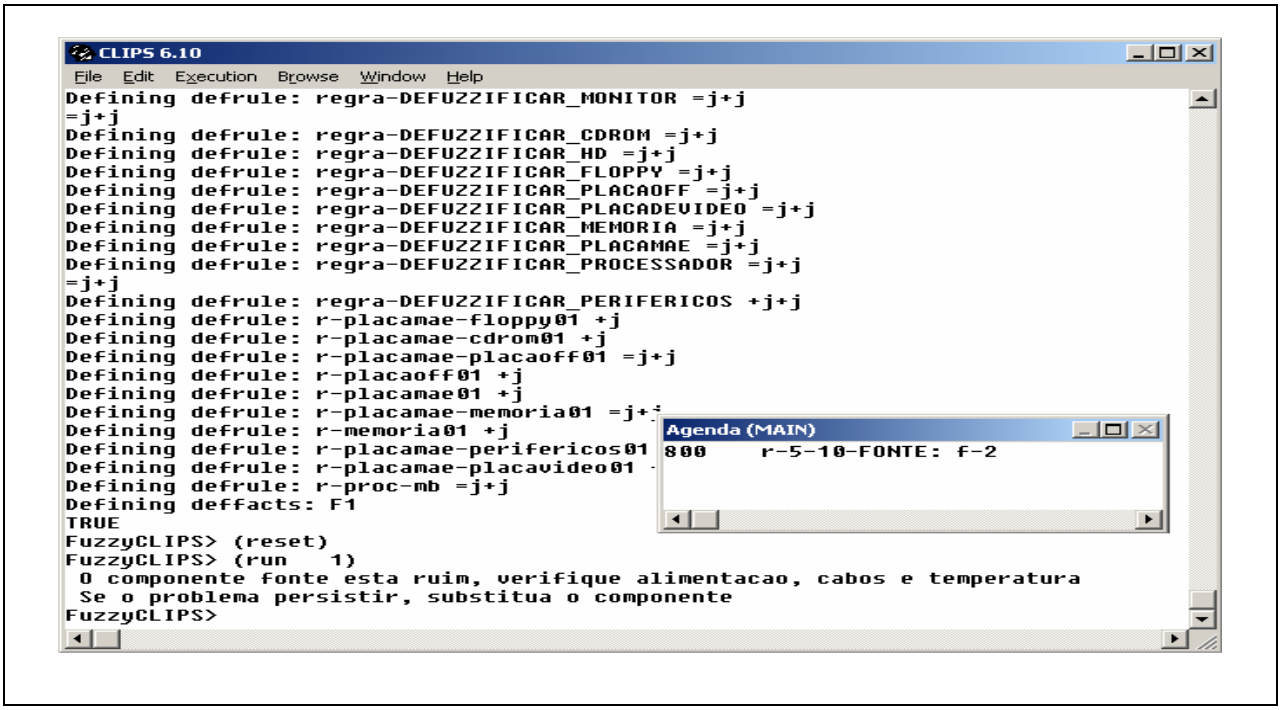


Figura 30. Interface Teste01

A partir dos fatos informados, é realizada uma inferência pelo método Mandani e ao final pelo método do centro de gravidade para efetuar a defuzzyficação e informar o resultado, onde o resultado mostra que o componente fonte, tem um coeficiente de certeza de 87% de estar com defeito. Com essa informação o técnico substitui a fonte, e atualiza os fatos de entrada, a fim de verificar se resolveu o problema, a Figura 31 exibe como a inferência é feita.

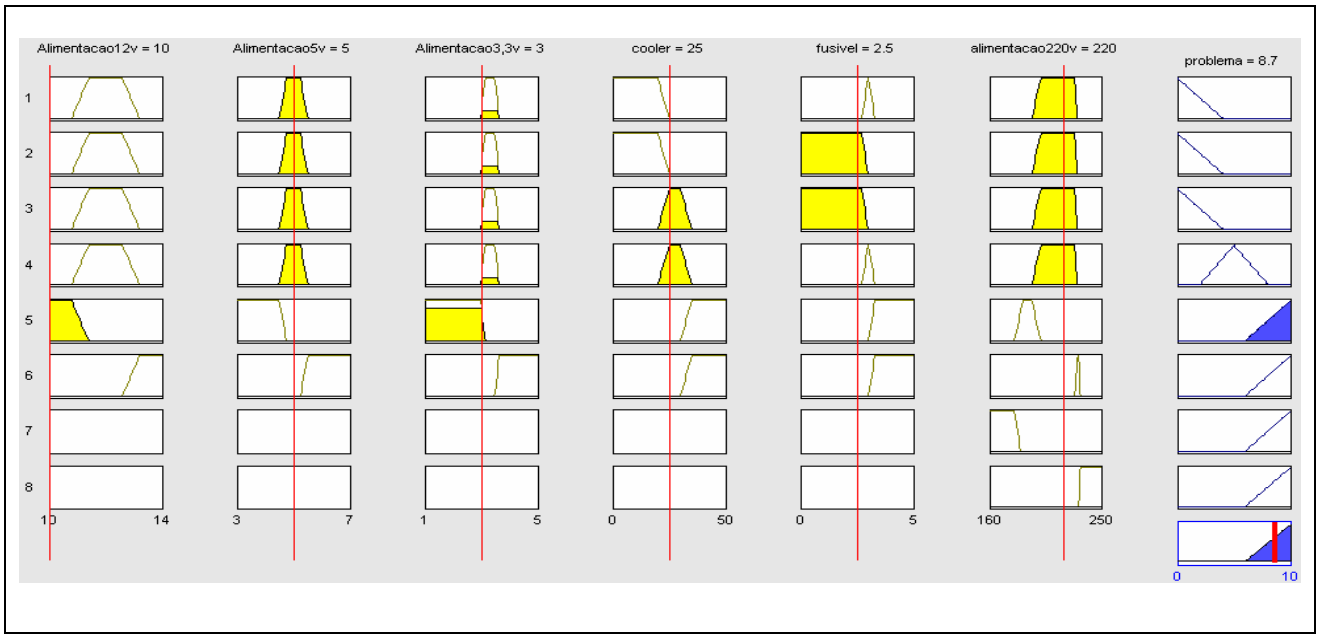


Figura 31. Teste01 de inferência da Fonte.

4.2 Teste 02

Um segundo teste realizado, foi referente a um chamado onde o usuário afirma que após algum tempo de utilização o computador trava. O técnico confirma o problema no local e após efetuar alguns testes verifica que o *cooler* da fonte de alimentação não esta funcionando. O técnico então efetua a substituição do componente e realiza um novo teste na máquina, e verifica que o problema não permanece (Tabela 4).

Tabela 4. Chamado 02

Descrição do problema	Fato	Ação Corretiva
... “computador está travando”...	Cooler fonte do computador está com problema.	Cooler da fonte trocado.

O técnico ao executar o sistema irá entrar com os fatos referentes a alimentação do computador, saída 12v, 5v, e 3,3v assim como a temperatura da fonte e corrente que esta passando pela fonte no momento, o sistema então irá avaliar as regras r-1-10-FONTE a r-5-10-FONTE, onde que baseado nos fatos informados irá disparar a regra r-5-10-FONTE, exibindo a mensagem " O componente fonte esta ruim, verifique alimentação, cabos e temperatura. Se o problema persistir, substitua o componente " como pode ser visto na Figura 32.

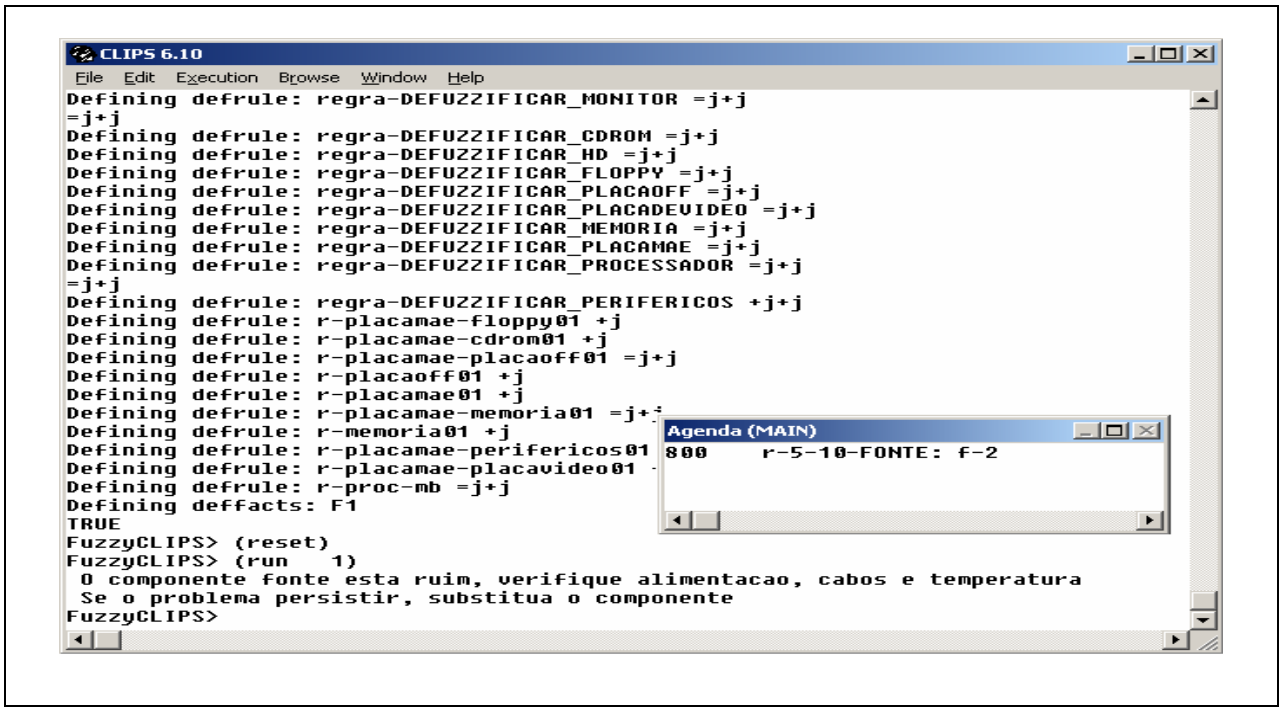


Figura 32. Interface Teste02

A Figura 33 exibe onde esta sendo realizada a inferência dos fatos e a defuzzyficação dos resultados, mostrado ao final um fator de certeza de 67% de a fonte estar com problema.

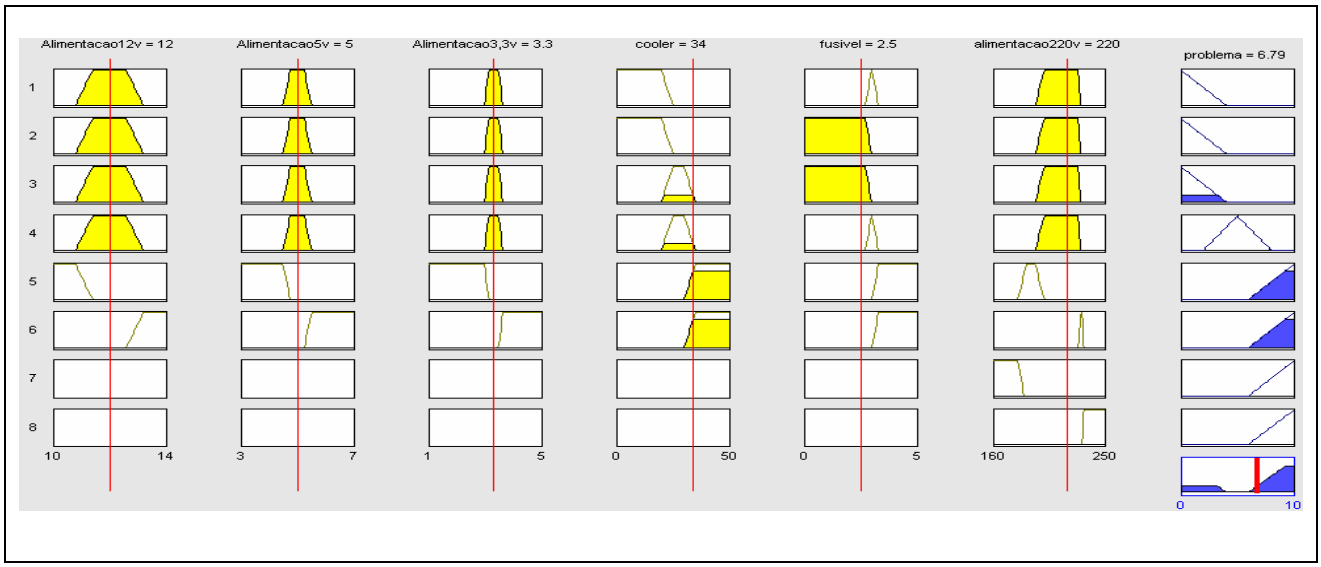


Figura 33. Teste02 da Fonte.

4.3 Teste 03

O terceiro teste realizado foi em relação a um chamado onde o usuário informa que o sistema operacional não está iniciando, o técnico então efetua uma bateria de testes e verifica que o cabo flat do HD está com problema, após realizar a troca, o microcomputador é testado novamente e verifica que o mesmo voltou a funcionar normalmente (Tabela 5).

Tabela 5. Chamado 03

Descrição do problema	Fato	Ação Corretiva
... “o computador não está ligando, este problema acontece freqüentemente devido a forma como transportamos o computador”...	Computador não inicializa do windows.	Cabo flat do HD com defeito. Substituído

Executando o sistema o primeiro teste solicitado são os referentes a alimentação do computador, descrito no exemplo anterior, que após a entrada de fatos, é disparada a regra r-2-10-FONTE, e posteriormente a regra de defuzzyficação, exibindo na tela a mensagem, “A fonte do computador tem: 1.3333 de chance de estar com problema” passando assim para o teste do próximo componente, o monitor, que após informar os fatos de entrada é disparada a regra r-1-4-MONITOR, para então fazer a defuzzyficação do resultado, exibindo a mensagem “O monitor tem: 8,33 de chance de estar OK”, o próximo teste é referente ao CD-ROM, que também ao inserir todos os fatos referentes ao componente, é disparada a regra r-1-3-CDROM, e logo após a regra de defuzzyficação, que exibe a mensagem “O CD-ROM tem: 8,3 de chance de estar OK”.

O próximo teste é o do HD. O sistema solicita ao técnico que verifique se o HD está sendo identificado pela BIOS do computador, o técnico então informa que não está sendo reconhecido, o sistema pergunta ao técnico como está a situação dos cabos de força e o cabo flat e verificar a posição do *jumper*, com base nos fatos informados, as regras r-1-2-HD e r-2-2-HD são avaliadas e a regra r-2-2-HD é disparada, exibindo a mensagem “Provavelmente o HD apresenta algum problema, verifique cabos e *jumpers*. Se o problema Persistir substitua o componente” como mostra a Figura 34.

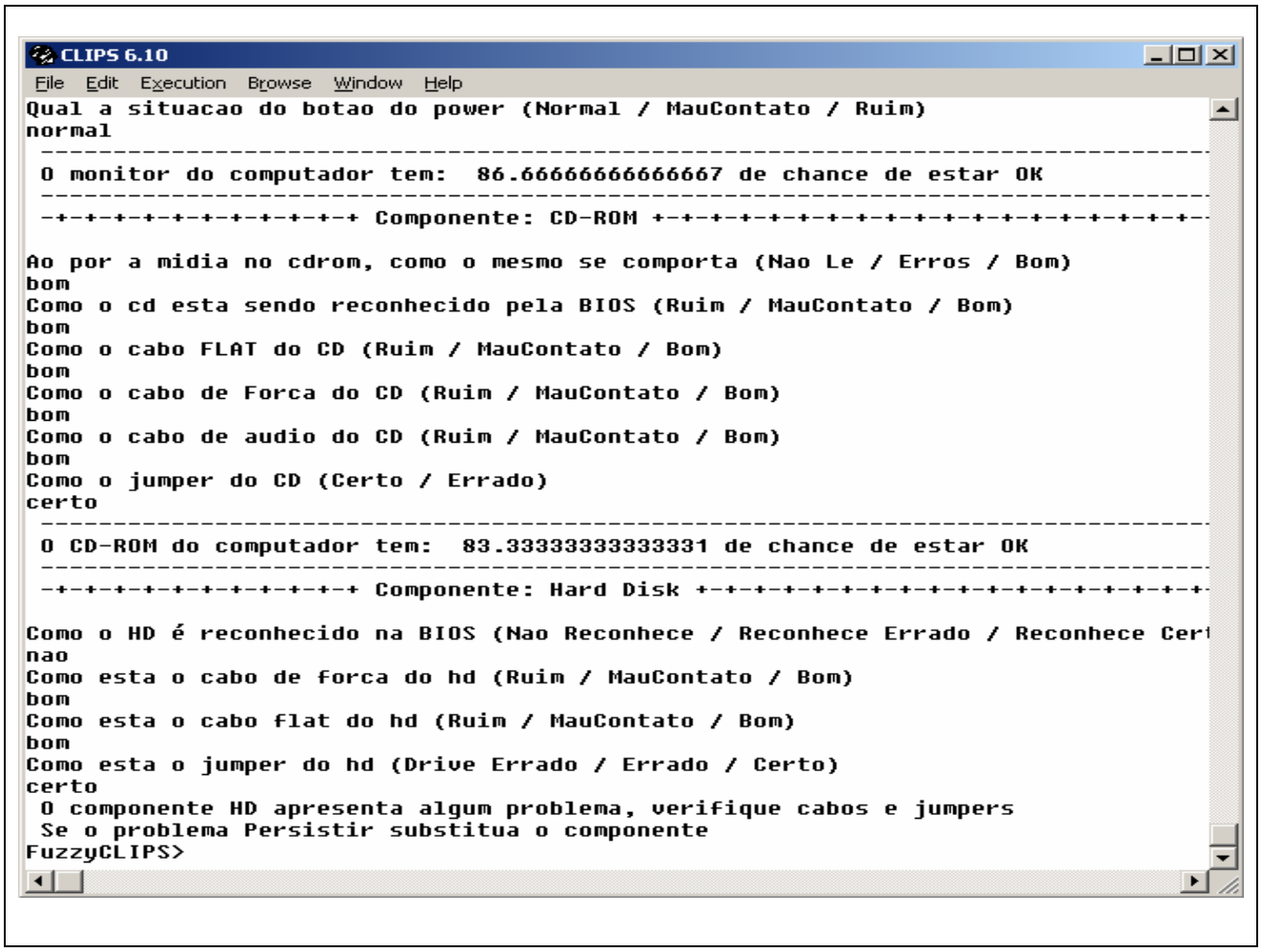

CLIPS 6.10
File Edit Execution Browse Window Help
Qual a situacao do botao do power (Normal / MauContato / Ruim)
normal
O monitor do computador tem: 86.66666666666667 de chance de estar OK
Componente: CD-ROM
Ao por a midia no cdrom, como o mesmo se comporta (Nao Le / Erros / Bom)
bom
Como o cd esta sendo reconhecido pela BIOS (Ruim / MauContato / Bom)
bom
Como o cabo FLAT do CD (Ruim / MauContato / Bom)
bom
Como o cabo de Forca do CD (Ruim / MauContato / Bom)
bom
Como o cabo de audio do CD (Ruim / MauContato / Bom)
bom
Como o jumper do CD (Certo / Errado)
certo
O CD-ROM do computador tem: 83.33333333333331 de chance de estar OK
Componente: Hard Disk
Como o HD é reconhecido na BIOS (Nao Reconhece / Reconhece Errado / Reconhece Cer
nao
Como esta o cabo de forca do hd (Ruim / MauContato / Bom)
bom
Como esta o cabo flat do hd (Ruim / MauContato / Bom)
bom
Como esta o jumper do hd (Drive Errado / Errado / Certo)
certo
O componente HD apresenta algum problema, verifique cabos e jumpers
Se o problema Persistir substitua o componente
FuzzyCLIPS>

Figura 34. Interface Teste03

Na Figura 35 é exibida a inferência dos 4 fatos de entrada e a defuzzyficação efetuada, resultando em um coeficiente de certeza de 13% do HD estar funcionando corretamente. Após a troca dos cabos e verificado a posição do *jumper*, o técnico efetua um novo teste no microcomputador a fim de verificar se o problema foi resolvido, a Figura 32 exibe como a inferência é feita.

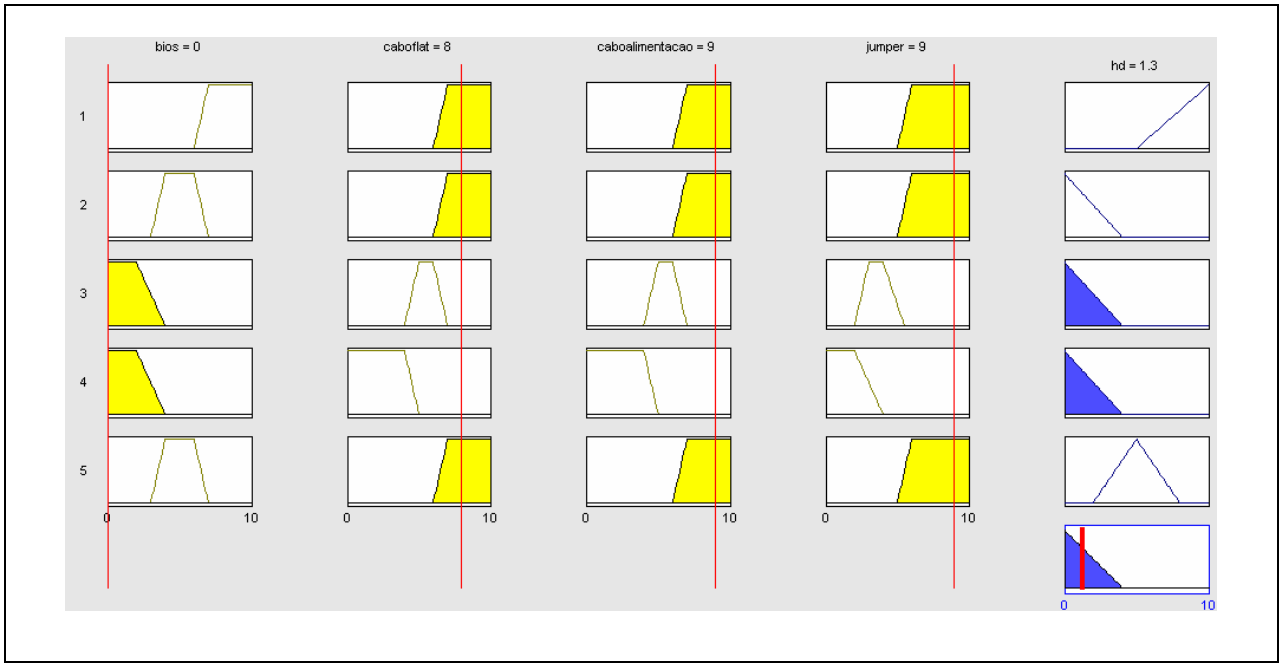


Figura 35. Teste03 do HD.

4.4 Teste 04

No quarto teste o usuário informa "... Desktop está desconfigurado, os ícones estão num tamanho muito grande...", ao chegar no local o técnico constata que o sistema operacional não esta mantendo as configurações de vídeo, o técnico então efetua a instalação de uma placa de vídeo *off-board* que resolve o problema (Tabela 6).

Tabela 6. Chamado 04

Descrição do problema	Fato	Ação Corretiva
... "DeskTop está desconfigurado, os ícones estão num tamanho muito grande" ...	micro perdendo configuração de vídeo.	colocado placa de vídeo off-board.

Quando o técnico executa o programa, após passar pelos testes, de fonte, monitor, CD-ROM, HD, FLOPPY, Placas *Off-board*, o sistema chega no componente Placa de vídeo, o sistema solicita ao técnico que se possível mudar o *slot* da placa de vídeo, o técnico informa que "não", posteriormente o sistema pede para testar outra placa de vídeo, é realizado a inferência e as regras r-1-3-PLACADEVIDEO a r-3-3-PLACADEVIDEO são avaliadas, onde o sistema dispara a regra r-3-3-PLACADEVIDEO, exibindo a mensagem "O componente placa de vídeo esta ruim, troque de *slot*, verifique possíveis mau contatos. Se o problema persistir substitua o componente" como pode ser visto na Figura 36.

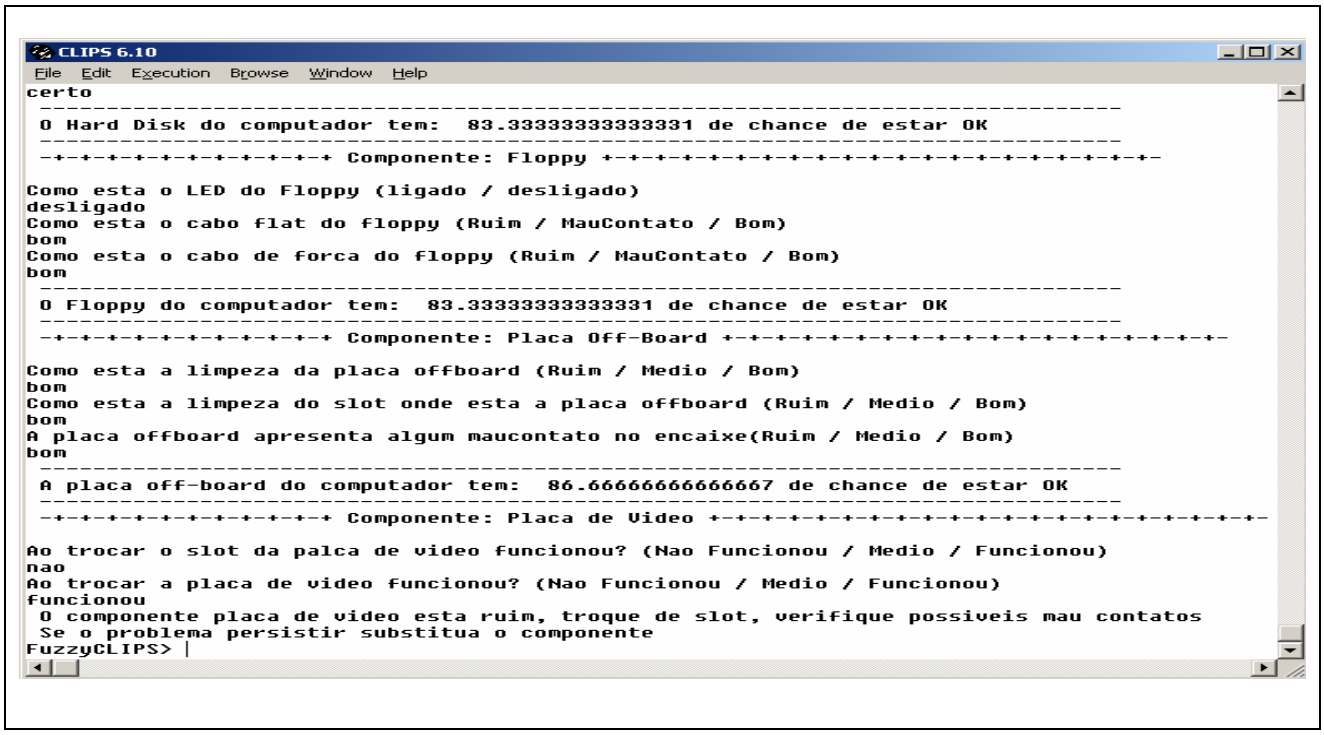

CLIPS 6.10
File Edit Execution Browse Window Help
certo
O Hard Disk do computador tem: 83.33333333333331 de chance de estar OK
Componente: Floppy
Como esta o LED do Floppy (ligado / desligado)
desligado
Como esta o cabo flat do floppy (Ruim / MauContato / Bom)
bom
Como esta o cabo de forca do floppy (Ruim / MauContato / Bom)
bom
O Floppy do computador tem: 83.33333333333331 de chance de estar OK
Componente: Placa Off-Board
Como esta a limpeza da placa offboard (Ruim / Medio / Bom)
bom
Como esta a limpeza do slot onde esta a placa offboard (Ruim / Medio / Bom)
bom
A placa offboard apresenta algum maucontato no encaixe(Ruim / Medio / Bom)
bom
A placa off-board do computador tem: 86.66666666666667 de chance de estar OK
Componente: Placa de Video
Ao trocar o slot da palca de video funcionou? (Nao Funcionou / Medio / Funcionou)
nao
Ao trocar a placa de video funcionou? (Nao Funcionou / Medio / Funcionou)
funcionou
O componente placa de video esta ruim, troque de slot, verifique possiveis mau contatos
Se o problema persistir substitua o componente
FuzzyCLIPS>

Figura 36. Interface Teste04

A Figura 37 exibe que após defuzzyficar o resultado da inferência exibe um fator de certeza de 13% de chance da placa de vídeo estar funcionando corretamente.

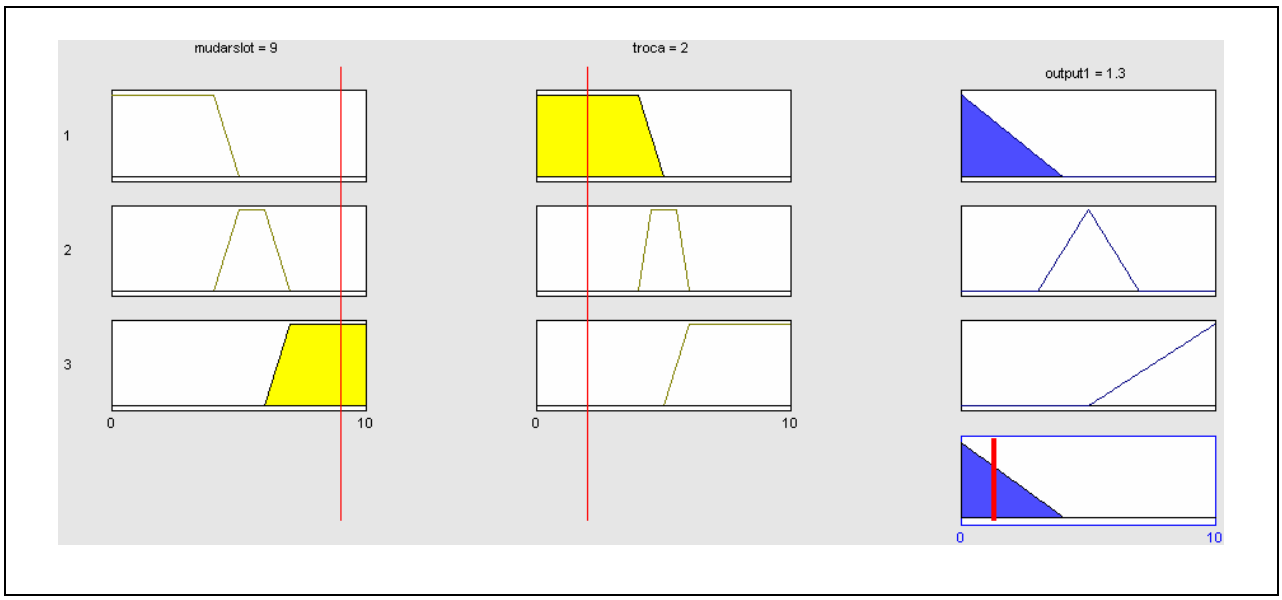


Figura 37. Teste04 da Placa de Video.

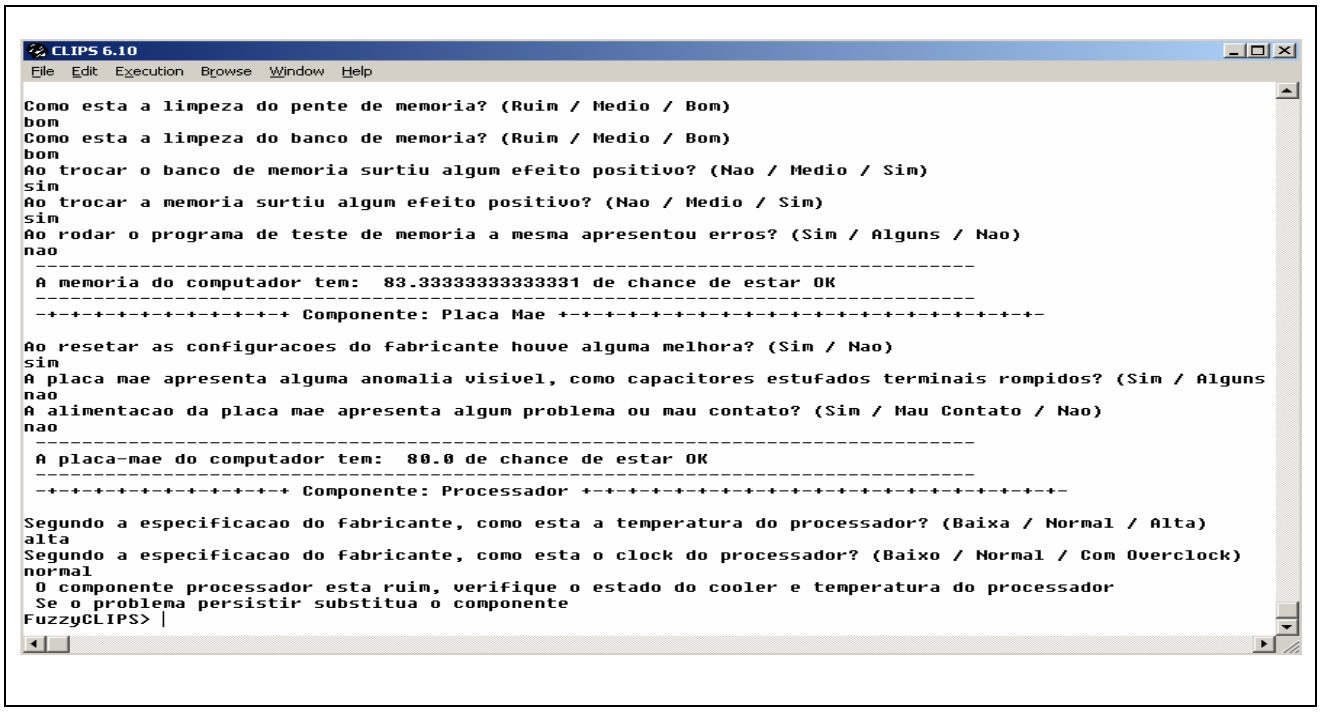
4.5 Teste 05

Outro teste realizado foi referente a um chamado onde o usuário comunica que o microcomputador esta travando. O técnico confirma o problema, e após alguns testes constata que o cooler do processador não está funcionando, ocasionando assim o aquecimento do mesmo (Tabela 7).

Tabela 7. Chamado 05

Descrição do problema	Fato	Ação Corretiva
... “computador desliga sozinho.”...	Computador travando	Ligação do cooler do processador no pino da fonte.

Após responder perguntas sobre a fonte, monitor, CD-ROM, HD, FLOPPY, Placas *Off-board*, Placa de Vídeo, Memória e Placa Mãe, o sistema então pergunta se a temperatura do processador está dentro dos limites especificados pelo fabricante, o técnico informa que a temperatura esta mais alta que o normal, o sistema em seguida pergunta se o processador esta operando na freqüência especificada também pelo fabricante, o técnico informa que sim. Assim, regras r-1-3-PROCESSADOR a r-3-3-PROCESSADOR, são avaliadas e a regra r-3-3-PROCESSADOR então é disparada, exibindo a mensagem “O componente processador esta ruim, verifique o estado do cooler e temperatura do processador. Se o problema persistir substitua o componente” como pode ser observado na Figura 38.


CLIPS 6.10
File Edit Execution Browse Window Help
Como esta a limpeza do pente de memoria? (Ruim / Medio / Bom)
bom
Como esta a limpeza do banco de memoria? (Ruim / Medio / Bom)
bom
Ao trocar o banco de memoria surtiu algum efeito positivo? (Nao / Medio / Sim)
sim
Ao trocar a memoria surtiu algum efeito positivo? (Nao / Medio / Sim)
sim
Ao rodar o programa de teste de memoria a mesma apresentou erros? (Sim / Alguns / Nao)
nao
A memoria do computador tem: 83.33333333333331 de chance de estar OK
-+-+-+-+-+-+-+-+-+-+ Componente: Placa Mae +-+-+-+-+-+-+-+-+-+-+

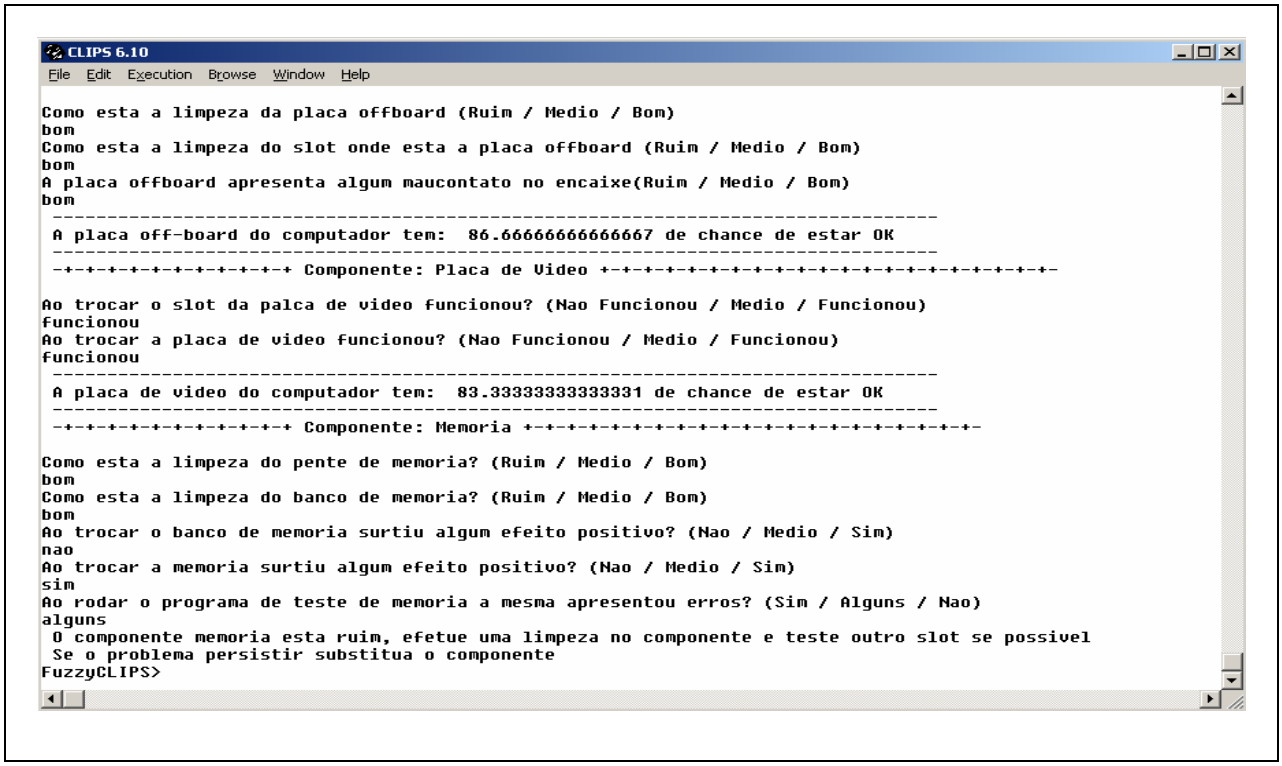
4.6 Teste 06

Um dos testes realizados foi um que o usuário informa que o micro apresenta travamento sem causa aparente, o técnico após alguns testes, constata que a memória apresenta erros, logo a mesma é substituída e o microcomputador volta a operar normalmente (Tabela 8).

Tabela 8. Chamado 06

Descrição do problema	Fato	Ação Corretiva
... “maquina apresenta travamento”...	Memória danificada	Feito a substituição da memória

O sistema então é executado, e após fazer perguntas sobre fonte, monitor, CD-ROM, HD, FLOPPY, Placas *Off-board*, Placa de Vídeo, e o sistema informar que os componentes estão “Ok”, é perguntado ao técnico como esta a limpeza do pente de memória, se apresentando sinais de oxidação em seus terminais, o técnico responde que esta ok, o sistema pergunta se o banco de memória na placa mãe esta sujo, o técnico informa que não, pergunta se ao trocar o banco o problema continuou, o técnico informa que sim, se ao trocar a memória o problema permanece o técnico informa que não e por fim se ao rodar um software de teste de memória, a memória apresenta erros, o técnico informa que sim. Com base nas respostas que representa os fatos de entrada, as regras r-1-3-MEMORIA a r-2-3-MEMORIA então são avaliadas e a regra r-2-3-MEMORIA é disparada, informando que a “O componente memória esta ruim, efetue uma limpeza no componente e teste outro *slot* se possível. Se o problema persistir substitua o componente” como pode ser visto na Figura 40.


CLIPS 6.10
File Edit Execution Browse Window Help
Como esta a limpeza da placa offboard (Ruim / Medio / Bom)
bom
Como esta a limpeza do slot onde esta a placa offboard (Ruim / Medio / Bom)
bom
A placa offboard apresenta algum maucontato no encaixe(Ruim / Medio / Bom)
bom
A placa off-board do computador tem: 86.66666666666667 de chance de estar OK
-+-+-+-+-+-+-+-+-+-+ Componente: Placa de Video +-+-+-+-+-+-+-+-+-+-+

5 CONCLUSÕES

O desenvolvimento teve com principal objetivo obter um conhecimento mais profundo na área de Inteligência Artificial em relação a Sistemas Especialistas e lógica *fuzzy*. O projeto teve início com um estudo da lógica *fuzzy* onde foi possível verificar em quais tipos de problemas ela pode ser aplicada, juntamente com um estudo mais profundo dos Sistemas Especialistas, pois a integração das duas técnicas trabalhando em conjunto, é a forma mais indicada para a solução do objetivo proposto. Foi possível demonstrar que o desenvolvimento de sistemas que utilizem variáveis lingüísticas e não precisas podem chegar em um resultado tão bom quanto os desenvolvidos com variáveis precisas.

O sistema não visa a substituição do especialista, e sim mais uma ferramenta para auxiliá-lo a “lembrar” de alguns procedimentos que possam passar despercebidos uma vez que não sejam tão comuns como os demais.

Para a implementação do projeto foi escolhida a Sh*ell* FUZZYCLIPS, por sua capacidade de lidar com sistemas especialistas e lógica *fuzzy*, porém possui uma deficiência em relação a interface com o usuário, que comparada a outras linguagens de alto nível, se mostra pouco atrativa para com o usuário.

As regras foram desenvolvidas baseadas nas entrevistas com os especialistas, que informaram quais problemas mais comuns encontrados, e qual a seqüência adotada para diagnosticar tais problemas. Após a implementação do projeto foram realizados testes utilizando a o sistema de registro de chamados da Universidade do Vale do Itajaí, UNIVALI, a fim de comparar os resultados do sistema com os procedimentos adotados pelos técnicos em ambiente real.

Após a realização de uma bateria de testes o sistema apresentou resultados positivos, uma vez que o diagnóstico e a solução proposta pelo sistema coincidiam com as do ambiente real. Assim foi demonstrado que a utilização de variáveis lingüísticas se assemelha mais com a forma de pensar humana, diferente do método bivalente (sim ou não) dos computadores.

5.1 Trabalhos Futuros

O projeto possui pontos que podem ser ampliados ou aprofundados a fim de cobrir um universo ainda maior de problemas em micro-computadores. Sugestões de trabalhos futuros são:

Estudar uma forma de armazenar os resultados para um posterior relatório ou utilização dos dados em sistemas RBC.

Construir a integração com o Clips PHP a fim de disponibilizar este sistema via web para consultas rápidas sem a necessidade de instalação do FUZZYCLIPS na maquina.

Desenvolver uma interface mais amigável para o usuário utilizando a ferramenta JCLIPS por ser baseado em Java que é uma linguagem free.

Se aprofundar em problemas de software ou de configuração do sistema operacional, para o diagnóstico e uma gama ainda maior de problemas encontrados.

REFERÊNCIAS BIBLIOGRÁFICAS

ARAÚJO, Roberto da Silva; SIMAS, Etiene P. Lazzeris; RODRÍGUEZ, Antonio Gabriel; COSTA, Alex da. **Lógica Fuzzy: conceitos e aplicações.** Disponível em: <http://www.inf.unisinos.br/%7Ecazella/dss/fuzzy_relatorio.pdf >. Acesso em: 22 de Ago. 2007.

BITTENCOURT, G. **Inteligência Artificial** Ferramentas e Teorias. Ed. da UFSC, 1998, 362 p

BRANDÃO, Alessandro Castro. **Estendendo consultas SQL utilizando Lógica Fuzzy**. 2005. 60 f. (Monografia) Sistemas de Informação - Universidade Luterana do Brasil, Palmas, 2005.

CARVALHO, Roberto L; MONTEIRO, Sonia L.; GENARO, Sergio. **Engenharia do Conhecimento e Sistemas Especialistas**. Buenos Aires: Kapelusz, 1987. 100 p.

CASTILHO, E. ALVAREZ, E. **Expert system:** uncertainty and learning, the Alden Press. The Alden Press, Oxford, UK, 1991, 331 p.

CHORAFAS, Dimitris N. **Sistemas especialistas:** aplicações comerciais. São Paulo: MCGRAW-HILL, 1988, 254 p.

COSTA, Welbson; Siqueira; SILVA, Shirlly Christiany Macedo. **Aquisição de conhecimento:** o grande desafio na concepção de sistemas especialistas. Disponível em: <http://www.cefetrn.br/dpeq/holos/anterior/200509/pdf/037-046.pdf>. Acesso em: 22 de Ago. 2007.

DURKIN, Jonh. **Expert systems:** desing and development. Nova Iorque: Macmillian Publishing Company, 1994, 800 p.

FERNANDES, Anita Maria da Rocha. **Sistema especialista difuso aplicado ao processo de análise química qualitativa de amostras de minerais.** 1995. 146 f. **(**Dissertação) Mestrado em Ciências da Computação - Universidade Federal de Santa Catarina, Florianópolis, 1995.

GIARRATANO, Joseph C.; RILLEY, Gary. **Expert systems:** principles and programming, Boston: PWS Publishing Company, 1993. 597 p.

GONÇALVES, Vivian. **Um protótipo de sistema especialista para diagnosticar problemas em aparelhos de televisão**. (Monografia) Ciência da Computação - Universidade do Vale do Itajaí, São José, 1998, 167 p.

HARMON, Paul; KING, David. **Sistemas especialistas:** a inteligência artificial chega ao mercado. Edi. Campus, 1988.

LEVINE, Robert I. **Inteligência artificial e sistemas especialistas**, São Paulo: McGraw-Hill, 1988. 264 p.

MANCHINI, Daniella Patrícia; PAPPA, Gisele Lobo. **Sistemas Especialistas** Disponível em: <http://www.din.uem.br/ia/intelige/especialistas/especialistas/introdu.html>. Acesso em: 22 de Ago. 2007.

MASTELA, L. S. **Técnicas de aquisição de conhecimento para sistemas baseados em conhecimento**. 2004. 39 f. (Monografia) Ciências da computação - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.

ORTEGA, Neli Regina Siqueira. **Aplicação da Teoria de Conjuntos Fuzzy a problemas da biomedicina**. 2001. 166 f. (Tese) Doutorado em Física - Universidade São Paulo, São Paulo, 2001.

PASSOS, Emanuel Lopes. **Inteligência artificial e sistemas especialistas ao alcance de todos**. Rio de Janeiro: LCT, 1989, 195 p.

RABUSKE, Antonio R**. Inteligência Artificial**. Ed. da UFSC, 1995, 240 p.

REZENDE, Solange Oliveira. **Sistemas Inteligentes:** fundamentos e aplicações. São Paulo: Manole, 2002, 550 p.

RIBEIRO, Horacio da Cunha e Souza. **Introdução aos sistemas especialistas**. Rio de Janeiro: LTC, 1987, 142 p.

RUSSEL, Stuart; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Campus, 2004, 1021 p.

SANTOS, Rodrigo Eduardo dos; RODRIGUES, Santos e Joanir Luna. **Sistema Especialista para Diagnostico e Soluções de Problemas em Microcomputadores**, 2001. 48 f. (Monografia) Ciência da Computação - Universidade Federal de Mato Grosso do Sul, Campo Grande, 2001

SAVARIS, Silvana Valdemara Aparecida Michelotto, **Sistema especialista para primeiros socorros para cães**. 2002. 156 f. (Dissertação) Mestrado em Ciências da Computação-Universidade Federal de Santa Catarina, Florianópolis, 2002.

SILVA, João de Deus Mendes da; BESSANEZI, Rodney Carlos. Pontos Estacionários: I. Sistemas Dinâmicos P-fuzzy Unidimensionais. **BIOMATEMÁTICA 16**, UNICAMP, p. 65-88, 2006.

SCHNEIDER, Moti. **Fuzzy Expert System Tools**. Hoboken, NJ: John Wiley, 1996, 220 p.

SHAW, Ian S.; SIMÕES, Marcelo Godoy. **Controle e modelagem Fuzzy**, São Paulo: Edgard Blucher Ltda., 1999, 165 p.

SMART, Julian. **User Manual for wxCLIPS 1.63**, 1997, Disponível em: <http://scalab.uc3m.es/~docweb/ia/wxclips_manual/wxc_contents.html> Acesso em: 09 de Nov. 2007.

TEIVE, Raimundo Celeste Ghizoni. **Planejamento da expansão da transmissão de sistemas de energia elétrica utilizando sistemas especialistas**. 1991. 160 f. (Tese) Doutorado em Engenharia de Produção -Universidade Federal de Santa Catarina, Florianópolis, 1997.

TORRES, Gabriel, **Hardware curso completo**. Rio de Janeiro: Axcel Books 1999, 648 p.

ZADEH, L., **Fuzzy Sets as a basis for a Theory of Possibility, Fuzzy sets and systems**. Volume 100. Amsterdam: Elsevier North-Holland, Inc, 1978, 9-34 p

GLOSSÁRIO

Fusível	Dispositivo de proteção contra sobre-corrente em circuitos. Consiste de um filamento ou lâmina de um metal ou liga metálica de baixo ponto de fusão que se intercala em um ponto determinado de uma instalação elétrica para que se funda, por efeito Joule, quando a intensidade de corrente elétrica supere, devido a um curto-circuito ou sobrecarga, um determinado valor que poderia danificar a integridade dos condutores com o risco de incêndio ou destruição de outros elementos do circuito.
Cooler	Microventilador acoplado a uma placa dissipadora de calor, usado em diversos componentes para diminuir a sua temperatura.

APÊNDICES

A FLUXOGRAMA DE TESTES

A Tabela 9 demonstra a seqüência de testes adotada levando em consideração o problema do computador descrito.

A.1 PROBLEMAS MAIS COMUNS

Tabela 9. Sequência de testes

Módulo	Problemas
1	Computador não liga
2	Computador liga, mas não aparece nada no monitor.
3	Computador liga normalmente, porém emite ruídos, como “rangidos” e “cliques”.
3	Computador liga normalmente, porém cd-rom não funciona.
3	Computador liga normalmente, porém disquete não funciona.
3	Identifica os componentes, porém não carrega o sistema operacional.
3	Ao liga o computador, o mesmo emite “bips”, e nada aparece na tela.
4	Não identifica as placas *off-boards* no sistema operacional.
3	Não identifica algum dos *drivers*, disco rígido, cd-rom ou disquete.
9	Exibe mensagem na tela de teclado não encontrado, ou teclado não responde.
9	Mouse não funciona no sistema operacional.
1	Computador ao ligar, fica reiniciando.
1	Computador ao ligar, após algum tempo reinicia.
2	Computador ao ligar, o monitor apresenta padrões de caracteres não reconhecíveis.
3	*Drive* de disquete apresenta seu “led” ligado direto.
4	Computador esta ligado a um cabo de rede operante, porém ao entrar no sistema operacional a rede do computador não funciona e a configuração esta OK no Sistema Operacional.

Os módulos são enumerados de 1 a 9 como visto na Figura 42.

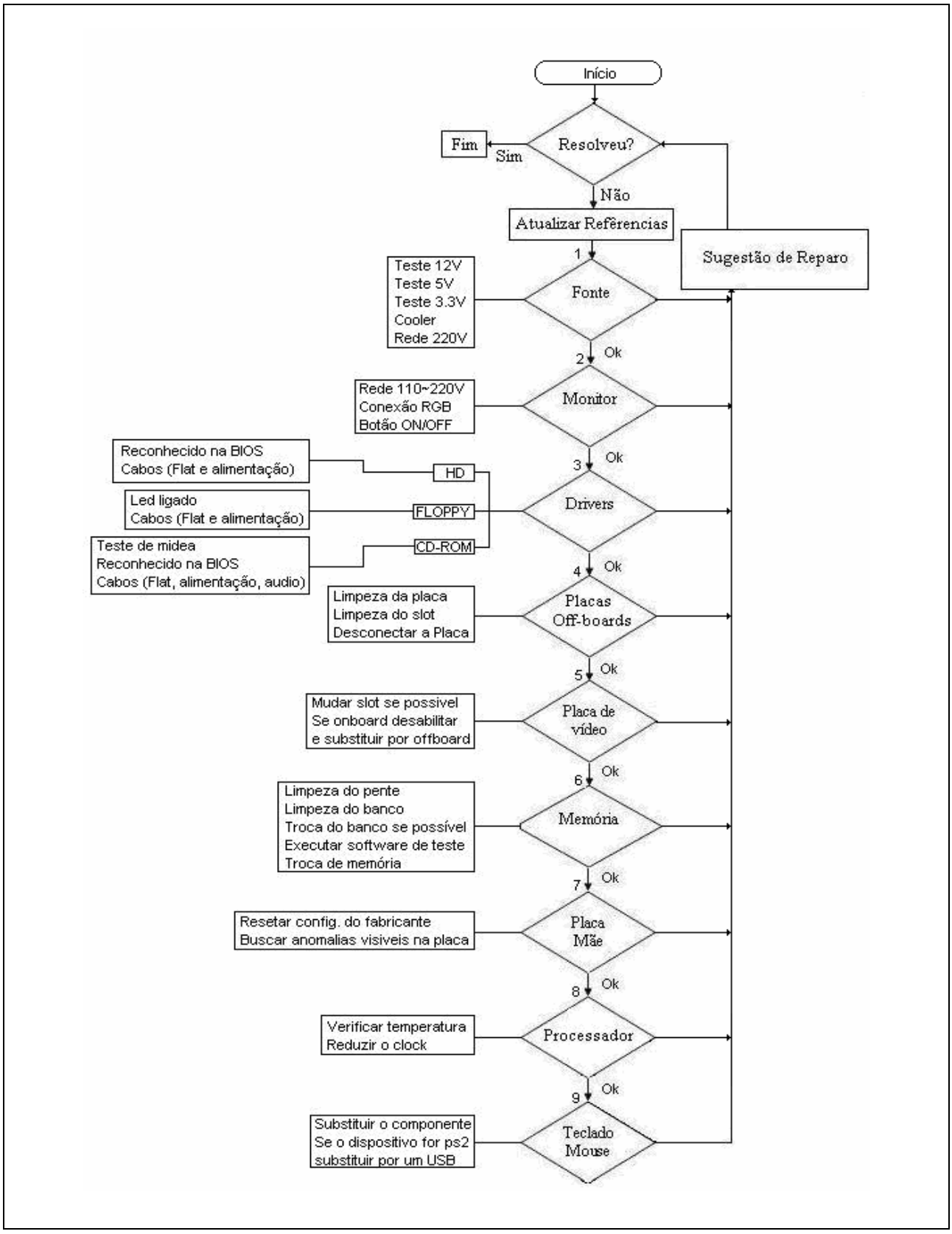


Figura 42. Fluxograma de testes baseado nos problemas